LAN Switching Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

CONTENTS

CHAPTER 1

Managed LAN Switch

The Managed LAN Switch feature enables the control of the four switch ports in Cisco 831, 836, and 837 routers. Each switch port is associated with a Fast Ethernet interface. The output of the **show controllers fastEthernet** commanddisplays the status of the selected switch port.

The Managed LAN Switch feature allows you to set and display the following parameters for each of the switch ports:

- Speed
- Duplex

It also allows you to display the link state of a switch port--that is, whether a device is connected to that port or not.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Managed LAN Switch

LAN Switching

A LAN is a high-speed, fault-tolerant data network that supplies connectivity to a group of computers, printers, and other devices that are in close proximity to each other, as in an office building, a school or a home. LANs offer computer users many advantages, including shared access to devices and applications, file exchange between connected users, and communication between users via electronic mail and other applications.

For more information about LAN switching, see the “LAN Switching” module of the *Internetworking Technology Handbook* .

How to Enable Managed LAN Switch

Enabling Managed LAN Switch

To enable Managed LAN Switch, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **duplex auto**
5. **speed auto**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet0/0`	Configures a Fast Ethernet interface and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**duplex auto** **Example:** `Router(config-if)# duplex auto`	Enables LAN switching on the selected port with duplex setting in auto mode.
Step 5	**speed auto** **Example:** `Router(config-if)# speed auto`	Enables LAN switching on the selected port with speed setting in auto mode.

Step 6	**end**
	Example:
	`Router(config-if)# end`
		Returns to privileged EXEC mode.

Verifying the Managed LAN Switch Configuration

To verify the Managed LAN Switch configuration, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **show controllers fastethernet** *number*
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show controllers fastethernet** *number* **Example:** `Router# show controllers fastethernet1`	Displays information about initialization block, transmit ring, receive ring, Fast Ethernet interface information, applicable MAC destination address and VLAN filtering tables, and errors for the Fast Ethernet controller chip. • Enter the port, connector, or interface card number.

Step 3	**end**
	Example:
	`Router(config-if)# end`
		Exits privileged EXEC mode.

Configuration Examples for Managed LAN Switch

Enabling the Managed LAN Switch Example

The following example shows the Managed LAN Switch configured with duplex set to auto and full, with speed set to auto and 100:

```
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
interface fastethernet1
no ip address
duplex auto
speed auto
!
interface fastethernet2
no ip address
duplex full  <---------------- duplex setting of port 2
speed 100 <-----------------   speed setting of port 2
!
interface fastethernet3
no ip address
shutdown   <--------------     shutting down port 3
duplex auto
speed auto
!
interface fastethernet4
no ip address
duplex auto
speed auto
!
```

Verifying the Managed LAN Switch Configuration Example

To verify the Managed LAN Switch configuration, enter the **show controllers fastethernet** <*1-4*> command in privileged EXEC mode. The following sample output shows the status of switch port 1.

```
Router# show controllers fastethernet1
!
Interface FastEthernet1   MARVELL 88E6052
Link is DOWN
Port is undergoing Negotiation or Link down
Speed :Not set, Duplex :Not set
!
Switch PHY Registers:
~~~~~~~~~~~~~~~~~~~~~
00 : 3100   01 : 7849   02 : 0141   03 : 0C1F   04 : 01E1
05 : 0000   06 : 0004   07 : 2001   08 : 0000   16 : 0130
17 : 0002   18 : 0000   19 : 0040   20 : 0000   21 : 0000
!
Switch Port Registers:
~~~~~~~~~~~~~~~~~~~~~~
Port Status Register       [00] : 0800
Switch Identifier Register [03] : 0520
Port Control Register      [04] : 007F
Rx Counter Register        [16] : 000A
Tx Counter Register        [17] : 0008
!
```

Additional References

The following sections provide references related to the Managed LAN Switch feature.

Related Documents

Related Topic	Document Title
IP LAN switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS LAN Switching Services Command Reference
LAN switching	"LAN Switching" module of the *Internetworking Technology Handbook*

Standards

Standards	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Managed LAN Switch

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 1: Feature Information for Managed LAN Switch

Feature Name	Releases	Feature Information
Managed LAN Switch	12.3(2)XC	This feature modifies the output of the **show controllers fastethernet** commandto show the status of switch port. The following command was modified: **show controllers fastethernet**

CHAPTER 2

Cisco HWIC-4ESW and HWIC-D-9ESW EtherSwitch Interface Cards

This document provides configuration tasks for the 4-port Cisco HWIC-4ESW and the 9-port Cisco HWIC-D-9ESW EtherSwitch high-speed WAN interface cards (HWICs) hardware feature supported on the Cisco 1800 (modular), Cisco 2800, and Cisco 3800 series Integrated Services Routers.

Cisco EtherSwitch HWICs are 10/100BASE-T Layer 2 Ethernet switches with Layer 3 routing capability. (Layer 3 routing is forwarded to the host and is not actually performed at the device.) Traffic between different VLANs on a device is routed through the device platform. Any one port on a Cisco EtherSwitch HWIC may be configured as a stacking port to link to another Cisco EtherSwitch HWIC or EtherSwitch network module in the same system. An optional power module can also be added to provide inline power for IP telephones. The HWIC-D-9ESW HWIC requires a double-wide card slot.

This hardware feature does not introduce any new or modified Cisco commands.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for EtherSwitch HWICs

- Configuration of IP routing. See the *IP Routing: Protocol-Independent Configuration Guide* for the Cisco software release you are using.
- Depending on your release, see the Cisco software documentation for the support of Cisco HWIC-4ESW and Cisco HWIC-D-9ESW.

Restrictions for EtherSwitch HWICs

- Not more than two EtherSwitch HWICs or network modules must be installed in a host device. Multiple EtherSwitch HWICs or network modules installed in a host device will not act independently of each other. They must be stacked, as they will not work otherwise.
- The ports of a Cisco EtherSwitch HWIC must not be connected to the Fast Ethernet/Gigabit onboard ports of the device.
- There must not be inline power on the ninth port (port 8) of the HWIC-D-9ESW card.
- There must not be Auto MDIX support on the ninth port (port 8) of the HWIC-D-9ESW card when either **speed** or **duplex** is not set to **auto**.
- There must not be support for online insertion/removal (OIR) of the EtherSwitch HWICs.
- When EtherSwitches have been installed and configured in a host device, OIR of the CompactFlash memory card in the device must not occur. OIR of the CompactFlash memory card will compromise the configuration of the EtherSwitches.
- VLAN Trunking Protocol (VTP) pruning is not supported.
- There is a limit of 200 secure MAC addresses per module that can be supported by an EtherSwitch HWIC.
- Maximum traffic for a secure MAC address is 8 Mb/s.

Prerequisites for Installing Two EtherSwitch Network Modules in a Single Chassis

A maximum of two EtherSwitch network modules can be installed in a single chassis. If two EtherSwitch network modules of any type are installed in the same chassis, the following configuration requirements must be met:

- Both EtherSwitch network modules must have an optional Gigabit Ethernet expansion board installed.

- An Ethernet crossover cable must be connected to the two EtherSwitch network modules using the optional Gigabit Ethernet expansion board ports.
- Intra-chassis stacking for the optional Gigabit Ethernet expansion board ports must be configured. For information about intra-chassis stacking configuration, see the “16- and 36-Port EtherSwitch Module for Cisco 2600 Series, Cisco 3600 Series, and Cisco 3700 Series” feature module.

Note Without this configuration and connection, duplications will occur in the VLAN databases, and unexpected packet handling may occur.

Information About EtherSwitch HWICs

VLANs

For conceptual information about VLANs, see the “VLANs” section of the EtherSwitch Network feature module.

Inline Power for Cisco IP Phones

For conceptual information about inline power for Cisco IP phones, see the “Inline Power for Cisco IP Phones” section of the EtherSwitch Network feature module.

Layer 2 Ethernet Switching

For conceptual information about Layer 2 Ethernet switching, see the “Layer 2 Ethernet Switching” section of the EtherSwitch Network feature module.

802.1x Authentication

For conceptual information about 802.1x authentication, see the “802.1x Authentication” section of the EtherSwitch Network feature module.

Spanning Tree Protocol

For conceptual information about Spanning Tree Protocol, see the “Using the Spanning Tree Protocol with the EtherSwitch Network Module” section of the EtherSwitch Network feature module.

Cisco Discovery Protocol

For conceptual information about Cisco Discovery Protocol, see the “Cisco Discovery Protocol” section of the EtherSwitch Network feature module.

Switched Port Analyzer

For conceptual information about a switched port analyzer, see the “Switched Port Analyzer” section of the EtherSwitch Network feature module.

IGMP Snooping

For conceptual information about Internet Group Management Protocol (IGMP) snooping, see the “IGMP Snooping” section of the EtherSwitch Network feature module.

Storm Control

For conceptual information about storm control, see the “Storm Control” section of the EtherSwitch Network feature module.

Intrachassis Stacking

For conceptual information about intrachassis stacking, see the ‘Intrachassis Stacking” section of the EtherSwitch Network feature module.

Fallback Bridging

For conceptual information about fallback bridging, see the “Fallback Bridging” section of the EtherSwitch Network feature module.

Default 802.1x Configuration

The table shows the default 802.1x configuration:

Table 2: Default 802.1x Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled.

Feature	Default Setting
RADIUS server • IP address • UDP authentication port • Key	• None specified. • 1645. • None specified.
Per-interface 802.1x enable state	Disabled (force-authorized). The port transmits and receives normal traffic without 802.1x-based authentication of the client.
Periodic reauthentication	Disabled.
Number of seconds between reauthentication attempts	3600 sec.
Quiet period	60 sec (period in seconds, that the device remains in a quiet state following a failed authentication exchange with the client).
Retransmission time	30 sec (period in seconds, that the device waits for a response to an EAP request/identity frame from the client before retransmitting the request).
Maximum retransmission number	2 (number of times that the device sends an EAP-request/identity frame before restarting the authentication process).
Multiple host support	Disabled.
Client timeout period	30 sec (period in seconds, that the device waits for a response before retransmitting the request to the client, when relaying a request from the authentication server to the client). This setting is not configurable.
Authentication server timeout period	30 sec (the period in seconds, that the device waits for a reply before retransmitting the response to the server, when relaying a response from the client to the authentication server). This setting is not configurable.

802.1x Configuration Guidelines

The 802.1x authentication configuration guidelines are as follows:

- When the 802.1x protocol is enabled, ports are authenticated before any other Layer 2 feature is enabled.

- The 802.1x protocol is supported on Layer 2 static-access ports, but it is not supported on the following port types:
    - Trunk port—If you try to enable 802.1x on a trunk port, an error message is displayed, and 802.1x is not enabled. If you try to change the mode of an 802.1x-enabled port to trunk, the port mode is not changed.
    - Switched Port Analyzer (SPAN) destination port—You can enable 802.1x on a port that is a SPAN destination port; however, 802.1x is disabled until the port is removed as a SPAN destination. You can enable 802.1x on a SPAN source port.

How to Configure EtherSwitch HWICs

Configuring VLANs

Adding a VLAN Instance

A total of 15 VLANs can be supported by an EtherSwitch HWIC.

Perform this task to configure a Fast Ethernet interface as Layer 2 access:

SUMMARY STEPS

1. **enable**
2. **vlan database**
3. **vlan** *vlan-id*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**vlan database** **Example:** `Device#` **vlan database**	Adds an ethernet VLAN and enters VLAN configuration mode.
Step 3	**vlan** *vlan-id*	Adds an Ethernet VLAN and enters VLAN configuration mode.

	Command or Action	Purpose
	Example: `Device(vlan)# vlan 1`	• Enter the VLAN number.

Step 4	**end**
	Example:
	`Device(vlan)# end`
		Updates the VLAN database, propagates it throughout the administrative domain, and returns to privileged EXEC mode.

Deleting a VLAN Instance from the Database

You cannot delete the default VLANs for the following media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.

Perform the following task to delete a VLAN from the database:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **no vlan** *vlan-id*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan** *vlan-id*	Adds an Ethernet VLAN.

	Command or Action	Purpose
	Example: `Device(config)# vlan 3`	• Enter the VLAN number.
Step 4	**no vlan** *vlan-id* **Example:** `Device(config-vlan)# no vlan 3`	Deletes an Ethernet VLAN. • Enter the VLAN number.

Step 5	**end**
	Example:
	`Device(config-vlan)# end`
		Updates the VLAN database, propagates it throughout the administrative domain, and returns to privileged EXEC mode.

Configuring VLAN Trunking Protocol

Note VTP pruning is not supported by EtherSwitch HWICs.

Configuring a VTP Server

When a device is in VTP server mode, you can change the VLAN configuration and propagate it throughout the network.

Perform this task to configure the device as a VTP server:

SUMMARY STEPS

1. **enable**
2. **vlan database**
3. **vtp server**
4. **vtp domain** *domain -name*
5. **vtp password** *password -value*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**vlan database** **Example:** `Device# vlan database`	Enters VLAN configuration mode.
Step 3	**vtp server** **Example:** `Device(vlan)# vtp server`	Configures the device as a VTP server.
Step 4	**vtp domain** *domain -name* **Example:** `Device(vlan)# vtp domain` *distantusers*	Defines the VTP domain name. • *domain name*- Enter the VTP domain name. Domain names can be a maximum of 32 characters.
Step 5	**vtp password** *password -value* **Example:** `Device(vlan)# vtp password` *password1*	(Optional) Sets a VTP domain password. • Specify a password. Passwords can be from 8 to 64 characters.

Step 6	**end**
	Example:
	`Device(vlan)# end`
		Updates the VLAN database, propagates it throughout the administrative domain, exits VLAN configuration mode, and returns to privileged EXEC mode.

Configuring a VTP Client

When a device is in a VTP client mode, you cannot change the VLAN configuration on the device. The client device receives VTP updates from a VTP server in the management domain and modifies its configuration accordingly.

Perform this task to configure the device as a VTP client:

SUMMARY STEPS

1. **enable**
2. **vlan database**
3. **vtp client**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**vlan database** **Example:** `Device#` **vlan database**	Adds an ethernet VLAN and enters VLAN configuration mode.
Step 3	**vtp client** **Example:** `Device(vlan)#` **vtp client**	Configures the device as a VTP client.
Step 4	**exit** **Example:** `Device(vlan)#` **exit**	Updates the VLAN database, propagates it throughout the administrative domain, exits VLAN configuration mode, and returns to privileged EXEC mode.

Disabling VTP (Transparent Mode)

When you configure the device in a VTP transparent mode, the VTP is disabled on the device. A VTP transparent device does not send VTP updates and does not act on VTP updates received from other devices.

Perform this task to disable VTP on the device.

SUMMARY STEPS

1. **enable**
2. **vlan database**
3. **vtp transparent**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**vlan database** **Example:** `Device# vlan database`	Adds an ethernet VLAN and enters VLAN configuration mode.
Step 3	**vtp transparent** **Example:** `Device(vlan)# vtp transparent`	Configures VTP transparent mode.

Step 4	**end**
	Example:
	`Device(vlan)# end`
		Updates the VLAN database, propagates it throughout the administrative domain, exits VLAN configuration mode, and returns to privileged EXEC mode.

Configuring Layer 2 Interfaces

Configuring a Range of Interfaces

Perform this task to configure a range of interfaces:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface range** {**macro** *macro-name* | **fastethernet** *interface-id* [**-** *interface-id*] | **vlan** *vlan-id*} [**,** **fastethernet** *interface-id* [- *interface-id*] | **vlan** *vlan-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface range** {**macro** *macro-name* | **fastethernet** *interface-id* [**-** *interface-id*] | **vlan** *vlan-id*} [**,** **fastethernet** *interface-id* [- *interface-id*] | **vlan** *vlan-id*] **Example:** `Device(config)# interface range FastEthernet 0/1/0 - 0/1/3`	Select the range of interfaces to be configured. • The space before the dash is required. For example, the command **interface range fastethernet**0/<*slot*>**/0 -**0/<*slot*>/**3** is valid; the command **interface range fastethernet**0/<*slot*>**/0-**0/<*slot*>**/3** is not valid. • You can enter one macro or up to five comma-separated ranges. • Comma-separated ranges can include both VLANs and physical interfaces. • You are not required to enter spaces before or after the comma. • The **interface range** command only supports VLAN interfaces that are configured with the **interface vlan** command.

Defining a Range Macro

Perform this task to define an interface range macro:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **define interface-range** *macro-name {* **fastethernet** *interface-id* [**-** *interface-id*] | {**vlan** *vlan-id* **-** *vlan-id*} | [**,** **fastethernet** *interface-id* [- *interface-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**define interface-range** *macro-name {* **fastethernet** *interface-id* [**-** *interface-id*] | {**vlan** *vlan-id* **-** *vlan-id*} | [**,** **fastethernet** *interface-id* [- *interface-id*] **Example:** `Device(config)# define interface-range first_three FastEthernet0/1/0 - 2`	Defines a range of macros. • Enter the macro name, along with the interface type and interface number, as appropriate.

Configuring Layer 2 Optional Interface Features

This section provides the following configuration information:

Configuring the Interface Speed

Perform this task to set the interface speed:

When configuring an interface speed, note these guidelines:

- If both ends of the line support auto negotiation, Cisco highly recommends the default auto negotiation settings.
- If one interface supports auto negotiation and the other end does not, configure interface speed on both interfaces; do not use the **auto** setting on the supported side.
- Both ends of the line need to be configured to the same setting; for example, hard-set or auto-negotiate. Mismatched settings are not supported.

Caution Changing the interface speed can shut down and reenable the interface during the reconfiguration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface fastethernet *interface-id*
4. **speed** {**10** | **100** | **1000** [**negotiate**] | **auto**[*speed-lis*t]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface fastethernet *interface-id* **Example:** `Device(config)# interface fastethernet 0/1/0`	Selects the interface to be configured and enters interface configuration mode. • Enter the interface number.

Step 4	**speed** {**10** | **100** | **1000** [**negotiate**] | **auto**[*speed-lis*t]}
	Example:
	`Device(config-if)# speed 100`
		Configures the speed for the interface.
		• Enter the desired speed.

What to Do Next

Note If you set the interface speed to auto on a 10/100-Mbps Ethernet interface, both speed and duplex are automatically negotiated.

Configuring the Interface Duplex Mode

Perform the following steps to set the duplex mode of a Fast Ethernet interface:

When configuring an interface duplex mode, note these guidelines:

- If both ends of the line support auto negotiation, Cisco highly recommends the default auto negotiation settings.
- If one interface supports auto negotiation and the other end does not, configure duplex speed on both interfaces; do not use the **auto** setting on the supported side.
- Both ends of the line need to be configured to the same setting, for example, hard-set or auto-negotiate. Mismatched settings are not supported.

⚠ **Caution** Changing the interface duplex mode configuration can shut down and reenable the interface during the reconfiguration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface fastethernet *interface-id*
4. **duplex** [**auto** | **full** | **half**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface fastethernet *interface-id* **Example:** `Device(config)# interface fastethernet 0/1/0`	Selects the interface to be configured. • Enter the interface number.

	Command or Action	Purpose
Step 4	**duplex** [**auto** | **full** | **half**] **Example:** `Device(config-if)# duplex auto`	Sets the duplex mode of the interface.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

What to Do Next

Note If you set the port speed to auto on a 10/100-Mbps Ethernet interface, both speed and duplex are automatically negotiated. You cannot change the duplex mode of auto negotiation interfaces.

Configuring a Description for an Interface

You can add a description of an interface to help you remember its function. The description appears in the output of the following commands: **show configuration**, **show running-config**, and **show interfaces**.

Use the **description** command to add a description for an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface fastethernet *interface-id*
4. **description** *string*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface fastethernet *interface-id* **Example:** `Device(config)# interface fastethernet 0/1/0`	Selects the interface to be configured and enters interface configuration mode. - Enter the interface number.
Step 4	**description** *string* **Example:** `Device(config-if)# description newinterface`	Adds a description for the interface. - Enter a description for the interface.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

Configuring a Fast Ethernet Interface as a Layer 2 Trunk

Perform the following task to configure a Fast Ethernet interface as a Layer 2 trunk.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface fastethernet *interface-id*
4. **shutdown**
5. switchport **mode trunk**
6. switchport **trunk native vlan** *vlan-number*
7. switchport **trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan1*[**,***vlan*[**,***vlan*[**,**...]]
8. **no shutdown**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface fastethernet *interface-id* **Example:** `Device(config)#` **interface fastethernet 0/1/0**	Selects the interface to be configured and enters interface configuration mode. • Enter the interface number.
Step 4	**shutdown** **Example:** `Device(config-if)# shutdown`	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 5	switchport **mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Configures the interface as a Layer 2 trunk. **Note** Encapsulation is always dot1q.
Step 6	switchport **trunk native vlan** *vlan-number* **Example:** `Device(config-if)# switchport trunk native vlan 1`	(Optional) For 802.1Q trunks, specifies the native VLAN.
Step 7	switchport **trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan1*[**,***vlan*[**,***vlan*[**,**...]] **Example:** `Device(config-if)# switchport trunk allowed vlan add vlan1, vlan2, vlan3`	(Optional) Configures the list of VLANs allowed on the trunk. All VLANs are allowed by default. You cannot remove any of the default VLANs from a trunk.
Step 8	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Activates the interface. (Required only if you shut down the interface.)

	Command or Action	Purpose

Step 9	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

What to Do Next

Note Ports do not support Dynamic Trunk Protocol (DTP). Ensure that the neighboring device is set to a mode that will not send DTP.

Configuring a Fast Ethernet Interface as Layer 2 Access

Perform the following task to configure a Fast Ethernet interface as Layer 2 access.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface fastethernet *interface-id*
4. **shutdown**
5. **switchport mode access**
6. **switchport access vlan** *vlan-number*
7. **no shutdown**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface fastethernet *interface-id* **Example:** `Device(config)# interface fastethernet 0/1/0`	Selects the interface to be configured and enters interface configuration mode. • Enter the interface number.
Step 4	**shutdown** **Example:** `Device(config-if)# shutdown`	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Configures the interface as a Layer 2 access.
Step 6	**switchport access vlan** *vlan-number* **Example:** `Device(config-if)# switchport access vlan 1`	For access ports, specifies the access VLAN. • Enter the VLAN number.
Step 7	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Activates the interface. • Required only if you shut down the interface.

Step 8	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

Configuring 802.1x Authentication

Enabling 802.1x Authentication

To enable 802.1x port-based authentication, you must enable Authentication, Authorization, and Accounting (AAA) and specify the authentication method list. A method list describes the sequence and authentication methods to be queried to authenticate a user.

The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is a successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle, the authentication process stops, and other authentication methods are not attempted.

For additional information about default 802.1x configuration, see “Default 802.1x Configuration” section.

Perform the following task to configure 802.1x port-based authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication dot1x** {**default** | *listname*} *method1* [*method2...*]
4. **interface** *interface-type interface-number*
5. **dot1x port-control auto**
6. **end**
7. **show dot1x**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication dot1x** {**default** | *listname*} *method1* [*method2...*] **Example:** `Device(config)# aaa authentication dot1x default newmethod`	Creates an 802.1x authentication method list. • To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword, followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces. • Enter at least one of these keywords: • **group radius—**Use the list of all RADIUS servers for authentication. • **none—**Use no authentication. The client is automatically authenticated without the device using the information supplied by the client.

	Command or Action	Purpose
Step 4	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/1/3`	Specifies the interface to be enabled for 802.1x authentication and enters interface configuration mode. • Enter the interface type and interface number.
Step 5	**dot1x port-control auto** **Example:** `Device(config-if)# dot1x port-control auto`	Enables 802.1x on the interface. • For feature interaction information with trunk, dynamic, dynamic-access, EtherChannel, secure, and SPAN ports, see the "802.1x Configuration Guidelines" section on page 19 .

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Step 7	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring the Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by their hostname or IP address, hostname and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authentication—the second host entry configured acts as the failover backup to the first one. The RADIUS host entries are tried in the order that they were configured.

Perform the following task to configure the RADIUS server parameters on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server host** {*hostname* | *ip-address*} **auth-port** *port-number* **key** *string*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server host** {*hostname* | *ip-address*} **auth-port** *port-number* **key** *string* **Example:** `Device(config)# radius-server host hostseven auth-port 75 key newauthority75`	Configures the RADIUS server parameters on the device. • For *hostname* | *ip-address,* specify the hostname or IP address of the remote RADIUS server. • For **auth-port** *port-number*, specify the UDP destination port for authentication requests. The default is 1645. • For **key** *string*, specify the authentication and encryption key used between the device and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server. **Note** Always configure the key as the last item in the **radius-server host** command syntax because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon. • If you want to use multiple RADIUS servers, repeat this command.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to Do Next

To delete the specified RADIUS server, use the **no radius-server host** {*hostname* | *ip-address*} global configuration command.

You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the device and the key string to be shared by both the server and the device. For more information, refer to the RADIUS server documentation.

Troubleshooting Tips

To delete the specified RADIUS server, use the **no radius server-host** { **hostname**|**ip-address**} global configuration command. You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and **radius-server key** commands in global configuration mode.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the device and the key string to be shared by both the server and the device. For more information, refer to the RADIUS server documentation.

Enabling Periodic Reauthentication

You can enable periodic 802.1x client reauthentication and specify how often it should occur. If you do not specify a time period before enabling reauthentication, the default time period between reauthentication attempts is 3600 seconds.

Automatic 802.1x client reauthentication is a global setting and cannot be set for clients connected to individual ports.

Perform the following task to enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x re-authentication**
4. **dot1x timeout re-authperiod** *seconds*
5. **end**
6. **show dot1x**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dot1x re-authentication** **Example:** `Device(config)# dot1x re-authentication`	Enables periodic reauthentication of the client. • Periodic reauthentication is disabled by default.
Step 4	**dot1x timeout re-authperiod** *seconds* **Example:** `Device(config)# dot1x timeout re-authperiod 120`	Sets the number of seconds between reauthentication attempts. • The range is from 1 to 4294967295; the default is 3600 seconds. • This command affects the behavior of the device only if periodic reauthentication is enabled

Step 5	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Changing the Quiet Period

If the device cannot authenticate the client, the device remains idle for a set period of time, and then tries again. The idle time is determined by the quiet-period value. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering smaller number than the default.

Perform the following task to change the quiet period.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x timeout quiet-period** *seconds*
4. **end**
5. **show dot1x**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**dot1x timeout quiet-period** *seconds* **Example:** `Device(config)# dot1x timeout quiet-period 120`	Sets the number of seconds that the device remains in the quiet state following a failed authentication exchange with the client. • The range is from 0 to 65535 seconds; the default is 60.

Step 4	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Step 5	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Changing the Device-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the device with an EAP-response/identity frame. If the device does not receive this response, it waits for a set period of time (known as the retransmission time), and then retransmits the frame.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Perform the following task to change the amount of time that the device waits for client notification.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x timeout tx-period** *seconds*
4. **end**
5. **show dot1x**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dot1x timeout tx-period** *seconds* **Example:** `Device(config)# dot1x timeout tx-period seconds`	Sets the number of seconds that the device waits for a response to an EAP-request/identity frame from the client before retransmitting the request. • The range is from 1 to 65535 seconds; the default is 30.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global interface configuration mode and returns to privileged EXEC mode.

Step 5	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Setting the Device-to-Client Frame-Retransmission Number

In addition to changing the device-to-client retransmission time, you can change the number of times that the device sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Perform the following task to set the device-to-client frame-retransmission number.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x max-req** *count*
4. **end**
5. **show dot1x**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dot1x max-req** *count* **Example:** `Device(config)# dot1x max-req 5`	Sets the number of times that the device sends an EAP-request/identity frame to the client before restarting the authentication process. • The range is from 1 to 10; the default is 2.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling Multiple Hosts

You can attach multiple hosts to a single 802.1x-enabled port. In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails, and an EAPOL-logoff message is received), all attached clients are denied access to the network.

Perform the following task to allow multiple hosts (clients) on an 802.1x-authorized port that has the **dot1x port-control** interface configuration command set to **auto**.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **dot1x multiple-hosts**
5. **end**
6. **show dot1x**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number*	Specifies the interface and enters interface configuration mode.

	Command or Action	Purpose
	Example: `Device(config)# interface fastethernet 0/1/2`	• Enter the interface type and interface number.
Step 4	**dot1x multiple-hosts** **Example:** `Device(config-if)# dot1x multiple-hosts`	Allows multiple hosts (clients) on an 802.1x-authorized port. • Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Step 6	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Resetting the 802.1x Configuration to the Default Values

You can reset the 802.1x configuration to the default values with a single command.

Perform the following task to reset the 802.1x configuration to the default values.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x default**
4. **end**
5. **show dot1x**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dot1x default** **Example:** `Device(config)# dot1x default`	Resets the configurable 802.1x parameters to the default values.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	**show dot1x** **Example:** `Device# show dot1x`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Displaying 802.1x Statistics and Status

To display 802.1x statistics for all interfaces, use the **show dot1x statistics** privileged EXEC command. To display 802.1x statistics for a specific interface, use the **show dot1x statistics interface** *interface-id* privileged EXEC command.

To display the 802.1x administrative and operational status for the device, use the **show dot1x** privileged EXEC command. To display the 802.1x administrative and operational status for a specific interface, use the **show dot1x interface** *interface-id* privileged EXEC command.

Configuring Spanning Tree

Enabling Spanning Tree Protocol

You can enable spanning tree protocol on a per-VLAN basis. The device maintains a separate instance of spanning tree for each VLAN except for which you disable spanning tree.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree vlan** *vlan-id*
4. **end**
5. **show spanning-tree vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree vlan** *vlan-id* **Example:** `Device(config)# spanning-tree vlan 200`	Enables spanning tree on a per-VLAN basis.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Step 5	**show spanning-tree vlan** *vlan-id* **Example:** `Device# show spanning-tree vlan 200`	Verifies spanning tree configuration.

Configuring Spanning Tree Port Priority

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **spanning-tree port-priority** *port-priority*
5. **spanning-tree vlan** *vlan-id* **port-priority** *port-priority*
6. **end**
7. **show spanning-tree interface fastethernet** *interface-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1/6`	Configures an interface and enters interface configuration mode.
Step 4	**spanning-tree port-priority** *port-priority* **Example:** `Device(config-if)# spanning-tree port-priority 8`	Configures the port priority for an interface.
Step 5	**spanning-tree vlan** *vlan-id* **port-priority** *port-priority* **Example:** `Device (config-if)# spanning-tree vlan vlan1 port-priority 12`	Configures the port priority for a VLAN.

Step 6	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show spanning-tree interface fastethernet** *interface-id* **Example:** `Device# show spanning-tree interface fastethernet 0/1/6`	(Optional) Saves your entries in the configuration file.

Configuring Spanning Tree Port Cost

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **spanning-tree cost** *port-cost*
5. **spanning-tree vlan** *vlan-id* **cost** *port-cost*
6. **end**
7. **show spanning-tree interface fastethernet** *interface-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1/6`	Configures an interface and enters interface configuration mode.
Step 4	**spanning-tree cost** *port-cost* **Example:** `Device(config-if)# spanning-tree cost 2000`	Configures the port cost for an interface.

	Command or Action	Purpose
Step 5	**spanning-tree vlan** *vlan-id* **cost** *port-cost* **Example:** `Device(config-if)# spanning-tree vlan 200 cost 2000`	Configures the VLAN port cost for an interface.

Step 6	**end**
	Example:
	`Device(config)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Step 7	**show spanning-tree interface fastethernet** *interface-id* **Example:** `Device# show spanning-tree interface fastethernet 0/1/6`	(Optional) Saves your entries in the configuration file.

Configuring the Bridge Priority of a VLAN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree vlan** *vlan-id* **priority** *bridge-priority*
4. **show spanning-tree vlan bridge**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree vlan** *vlan-id* **priority** *bridge-priority* **Example:** `Device(config)# spanning-tree vlan 200 priority 2`	Configures the bridge priority of a VLAN. The bridge priority value ranges from 0 to 65535. **Caution** Use the **spanning-tree vlan** *vlan-id* **root primary** command and the **spanning-tree vlan** *vlan-id* **root secondary** command to modify the bridge priority.

	Command or Action	Purpose
Step 4	**show spanning-tree vlan bridge** **Example:** `Device(config-if)# spanning-tree cost 200`	Verifies the bridge priority.

Configuring Hello Time

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree vlan** *vlan-id* **hello-time** *hello-time*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree vlan** *vlan-id* **hello-time** *hello-time* **Example:** `Device(config)# spanning-tree vlan 200 hello-time 5`	Configures the hello time for a VLAN.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuring the Forward Delay Time for a VLAN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree vlan** *vlan-id* **forward-time** *forward-time*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree vlan** *vlan-id* **forward-time** *forward-time* **Example:** `Device(config)# spanning-tree vlan 20 forward-time 5`	Configures the forward delay time for a VLAN.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuring the Maximum Aging Time for a VLAN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree vlan** *vlan-id* **max-age** *max-age*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree vlan** *vlan-id* **max-age** *max-age* **Example:** `Device(config)# spanning-tree vlan 200 max-age 30`	Configures the maximum aging time for a VLAN.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuring Spanning Tree Root Bridge

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spanning-tree vlan** *vlanid* **root primary** [**diameter** *hops* [**hello-time** *seconds*]]
4. **no spanning-tree vlan** *vlan-id*
5. **show spanning-tree vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**spanning-tree vlan** *vlanid* **root primary** [**diameter** *hops* [**hello-time** *seconds*]] **Example:** `Device(config)# spanning-tree vlan 200 root primary`	Configures a device as the root device.
Step 4	**no spanning-tree vlan** *vlan-id* **Example:** `Device(config)# no spanning-tree vlan 200 root primary`	Disables spanning tree on a per-VLAN basis.
Step 5	**show spanning-tree vlan** *vlan-id* **Example:** `Device(config)# show spanning-tree vlan 200`	Verifies spanning tree on a per-VLAN basis.

Configuring MAC Table Manipulation

Port security is implemented by providing the user with the option to secure a port by allowing only well-known MAC addresses to send in data traffic. Up to 200 secure MAC addresses per HWIC are supported.

Enabling Known MAC Address Traffic

Perform the following task to enable the MAC address secure option.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mac-address-table secure** *mac-address* **fastethernet** *interface-id* [**vlan** *vlan-id*]]
4. **end**
5. **show mac-address-table secure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac-address-table secure** *mac-address* **fastethernet** *interface-id* [**vlan** *vlan-id*]] **Example:** `Device(config)# mac-address-table secure 0000.0002.0001 fastethernet 0/1/1 vlan 2`	Secures the MAC address traffic on the port. • Enter the MAC address, the **fastethernet** keyword, the interface number, and any optional keywords and arguments as desired.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	**show mac-address-table secure** **Example:** `Device# show mac-address-table secure`	Verifies the configuration.

Creating a Static Entry in the MAC Address Table

Perform the following task to create a static entry in the MAC address table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mac-address-table static** *mac-address* **fastethernet** *interface-id* [**vlan** *vlan-id*]
4. **end**
5. **show mac-address-table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac-address-table static** *mac-address* **fastethernet** *interface-id* [**vlan** *vlan-id*] **Example:** `Device(config)# mac-address-table static 00ff.ff0d.2dc0 fastethernet 0/1/1`	Creates a static entry in the MAC address table. • When the *vlan-id* is not specified, VLAN 1 is taken by default.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show mac-address-table** **Example:** `Device# show mac-address-table`	Verifies the MAC address table.

Configuring and Verifying the Aging Timer

The aging timer may be configured from 16 seconds to 4080 seconds, in 16-second increments.

Perform this task to configure the aging timer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mac -address-table aging-tim e** *time*
4. **end**
5. **show mac-address-table aging-time**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac -address-table aging-tim e** *time* **Example:** `Device(config)# mac-address-table aging-time 4080`	Configures the MAC address aging timer age in seconds. • The range is from 0 to 10000 seconds.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show mac-address-table aging-time** **Example:** `Device# show mac-address-table aging-time`	Verifies the MAC address table.

Configuring Cisco Discovery Protocol

Enabling Cisco Discovery Protocol

To enable Cisco Discovery Protocol (CDP) globally, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cdp run**
4. **end**
5. **show cdp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cdp run** **Example:** `Device(config)# cdp run`	Enables CDP globally.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show cdp** **Example:** `Device# show cdp`	Verifies the CDP configuration.

Enabling CDP on an Interface

Perform this task to enable CDP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** {**ethernet** | **fastethernet**} *interface-id*
4. **cdp enable**
5. **end**
6. **show cdp interface** *interface-id*
7. **show cdp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** {**ethernet** | **fastethernet**} *interface-id* **Example:** `Device(config)# interface fastethernet 0/1/1`	Selects an interface and enters interface configuration mode. • Enter the interface number.
Step 4	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enables CDP globally.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

Step 6	**show cdp interface** *interface-id* **Example:** `Device# show cdp interface`	Verifies the CDP configuration on the interface.
Step 7	**show cdp neighbors** **Example:** `Device# show cdp neighbors`	Verifies the information about the neighboring equipment.

Monitoring and Maintaining CDP

Perform this task to monitor and maintain CDP on your device.

SUMMARY STEPS

1. **enable**
2. **clear cdp counter** s
3. **clear cdp table**
4. **show cdp**
5. **show cdp entry** *entry-name* [**protocol** | **version**]
6. **show cdp interface** *interface-id*
7. **show cdp neighbors** *interface-id* [**detail**]
8. **show cdp traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear cdp counter** s **Example:** `Device# clear cdp counters`	(Optional) Resets the traffic counters to zero.
Step 3	**clear cdp table** **Example:** `Device# clear cdp table`	(Optional) Deletes the CDP table of information about neighbors.
Step 4	**show cdp** **Example:** `Device# show cdp`	(Optional) Verifies global information such as frequency of transmissions and the holdtime for packets being transmitted.
Step 5	**show cdp entry** *entry-name* [**protocol** | **version**] **Example:** `Device# show cdp entry newentry`	(Optional) Verifies information about a specific neighbor. • The display can be limited to protocol or version information.
Step 6	**show cdp interface** *interface-id* **Example:** `Device# show cdp interface 0/1/1`	(Optional) Verifies information about interfaces on which CDP is enabled. • Enter the interface number.

	Command or Action	Purpose
Step 7	**show cdp neighbors** *interface-id* [**detail**] **Example:** `Device# show cdp neighbors 0/1/1`	(Optional) Verifies information about neighbors. • The display can be limited to neighbors on a specific interface and can be expanded to provide more detailed information.
Step 8	**show cdp traffic** **Example:** `Device# show cdp traffic`	(Optional) Verifies CDP counters, including the number of packets sent and received, and checksum errors.

Configuring the Switched Port Analyzer (SPAN)

Note An EtherSwitch HWIC supports only one SPAN session. Either Tx or both Tx and Rx monitoring is supported.

Configuring the SPAN Sources

Perform the following task to configure the source for a SPAN session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **monitor session 1** {**source interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-** | **rx** | **tx** | **both**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**monitor session 1** {**source interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-** | **rx** | **tx** | **both**] **Example:** `Device(config)# monitor session 1 source interface fastethernet 0/3/1`	Specifies the SPAN session (number 1), the source interfaces or VLANs, and the traffic direction to be monitored. • The example shows how to configure the SPAN session to monitor bidirectional traffic from source interface Fast Ethernet 0/3/1.

Configuring SPAN Destinations

Perform this task to configure the destination for a SPAN session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **monitor session** *session-id* {**destination** {**interface** *interface-id*} | {**vlan** *vlan-id*}} [, | - | **rx** | **tx** | **both**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**monitor session** *session-id* {**destination** {**interface** *interface-id*} | {**vlan** *vlan-id*}} [, | - | **rx** | **tx** | **both**]	Specifies the SPAN session (number 1), the source interfaces or VLANs, and the traffic direction to be monitored.

	Command or Action	Purpose
	Example: `Device(config)# monitor session 1 source interface fastethernet 0/3/1`	• The example shows how to configure the SPAN session to monitor bidirectional traffic from source interface Fast Ethernet 0/3/1.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode.

Configuring Power Management on the Interface

The HWICs can supply inline power to a Cisco 7960 IP phone, if necessary. The Cisco 7960 IP phone can also be connected to an AC power source and supply its own power to the voice circuit. When the Cisco 7960 IP phone is supplying its own power, an HWICs can forward IP voice traffic to and from the phone.

A detection mechanism on the HWIC determines whether the device is connected to a Cisco 7960 IP phone. If the device senses that there is no power on the circuit, the device supplies the power. If there is power on the circuit, the device does not supply it.

You can configure the device never to supply power to the Cisco 7960 IP phone and to disable the detection mechanism.

Follow these steps to manage the powering of the Cisco IP phones.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *interface-id*
4. **power inline** {**auto** | **never**}
5. **end**
6. **show power inline**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *interface-id* **Example:** `Device(config)# interface fastethernet 0/3/1`	Selects a particular Fast Ethernet interface for configuration, and enters interface configuration mode. • Enter the interface number.

Step 4	**power inline** {**auto** | **never**}
	Example:
	`Device(config-if)# power inline auto`
		Configures the port to supply inline power automatically to a Cisco IP phone.
		• Use **never** to permanently disable inline power on the port.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 6	**show power inline** **Example:** `Device# show power inline`	Displays power configuration on the ports.

Configuring IP Multicast Layer 3 Switching

Enabling IP Multicast Routing Globally

You must enable IP multicast routing globally before you can enable IP multicast Layer 3 switching on Layer 3 interfaces.

For complete information and procedures, see the following publications:

- *Protocol-Independent Configuration Guide*
- Cisco IOS IP Addressing Services Command Reference
- Cisco IOS IP Routing: Protocol-Independent Command Reference

Note See the Cisco command reference listing page for protocol-specific command references.

- Cisco IOS IP Multicast Command Reference

Perform the following task to enable IP multicast routing globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing globally.

Enabling IP Protocol-Independent Multicast (PIM) on Layer 3 Interfaces

You must enable protocol-independent multicast (PIM) on the Layer 3 interfaces before enabling IP multicast Layer 3 switching functions on those interfaces.

Perform this task to enable IP PIM on a Layer 3 interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan** *vlan-id*
4. **ip pim** {**dense-mode** | **sparse-mode** | **sparse-dense-mode**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface vlan** *vlan-id* **Example:** `Device(config)# interface vlan 1`	Selects the interface to be configured and enters interface configuration mode.

Step 4	**ip pim** {**dense-mode** | **sparse-mode** | **sparse-dense-mode**}
	Example:
	`Device(config-if)# ip pim sparse-dense mode`
		Enables IP PIM on a Layer 3 interface.

Verifying IP Multicast Layer 3 Hardware Switching Summary

Note The **show interface statistics** command does not verify hardware-switched packets; only packets switched by software are verified.

The **show ip pim interface count**command verifies the IP multicast Layer 3 switching enable state on IP PIM interfaces, and verifies the number of packets received and sent on the interface. Use the following **show** commands to verify IP multicast Layer 3 switching information for an IP PIM Layer 3 interface.

SUMMARY STEPS

1. Device# show ip pim interface count
2. Device# show ip mroute count
3. Device# show ip interface vlan 1

DETAILED STEPS

Step 1 Device# show ip pim interface count

Example:

```
State:* - Fast Switched, D - Distributed Fast Switched
      H - Hardware Switching Enabled
Address          Interface           FS  Mpackets In/Out
10.0.0.1         VLAN1               *   151/0
Device#
```

Step 2 Device# show ip mroute count

Example:

```
IP Multicast Statistics
5 routes using 2728 bytes of memory
4 groups, 0.25 average sources per group
Forwarding Counts:Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts:Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group:209.165.200.225 Source count:1, Packets forwarded: 0, Packets received: 66
  Source:10.0.0.2/32, Forwarding:0/0/0/0, Other:66/0/66
Group:209.165.200.226, Source count:0, Packets forwarded: 0, Packets received: 0
Group:209.165.200.227, Source count:0, Packets forwarded: 0, Packets received: 0
Group:209.165.200.228, Source count:0, Packets forwarded: 0, Packets received: 0
Device#
```

Note A negative counter means that the outgoing interface list of the corresponding entry is NULL, and this indicates that this flow is still active.

Step 3 Device# show ip interface vlan 1

Example:

```
Vlan1 is up, line protocol is up
  Internet address is 10.0.0.1/24
  Broadcast address is 209.165.201.1
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined:209.165.201.2 209.165.201.3 209.165.201.4 209.165.201.5
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Device Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
Device#
```

Verifying the IP Multicast Routing Table

Use the **show ip mroute** command to verify the IP multicast routing table:

```
show ip mroute 224.10.103.10
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel,
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags:H - Hardware switched, A - Assert winner
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode
(*, 209.165.201.2), 00:09:21/00:02:56, RP 0.0.0.0, flags:DC
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse-Dense, 00:09:21/00:00:00, H
Device#
```

Note The RPF-MFD flag indicates that the flow is completely hardware switched. The H flag indicates that the flow is hardware switched on the outgoing interface.

Configuring IGMP Snooping

Enabling or Disabling IGMP Snooping

By default, IGMP snooping is globally enabled on the EtherSwitch HWIC. When globally enabled or disabled, it is enabled or disabled in all existing VLAN interfaces. By default, IGMP snooping is enabled on all VLANs, but it can be enabled and disabled on a per-VLAN basis.

Global IGMP snooping overrides the per-VLAN IGMP snooping capability. If global snooping is disabled, you cannot enable VLAN snooping. If global snooping is enabled, you can enable or disable snooping on a VLAN basis.

Perform this task to globally enable IGMP snooping on the EtherSwitch HWIC.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping**
4.
5. **ip igmp snooping vlan** *vlan-id*
6. **end**
7. **show ip igmp snooping**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping** **Example:** `Device(config)# ip igmp snooping`	Globally enables IGMP snooping in all existing VLAN interfaces.
Step 4		
Step 5	**ip igmp snooping vlan** *vlan-id* **Example:** `Device(config)# ip igmp snooping vlan 100`	Globally enables IGMP snooping on a specific VLAN interface. • Enter the VLAN number.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 7	**show ip igmp snooping**	Displays snooping configuration.

	Command or Action	Purpose
	Example: `Device# show ip igmp snooping`	
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your configuration to the startup configuration.

Enabling IGMP Immediate-Leave Processing

When you enable IGMP Immediate-Leave processing, the EtherSwitch HWIC immediately removes a port from the IP multicast group when it detects an IGMP version 2 Leave message on that port. Immediate-Leave processing allows the device to remove an interface that sends a Leave message from the forwarding table without first sending out group-specific queries to the interface. You should use the Immediate-Leave feature only when there is only a single receiver present on every port in the VLAN.

Perform the following task to enable IGMP Immediate-Leave processing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **immediate-leave**
4. **end**
5. **show ip igmp snooping**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip igmp snooping vlan** *vlan-id* **immediate-leave** **Example:** `Device(config)# ip igmp snooping vlan 1 immediate-leave`	Enables IGMP Immediate-Leave processing on the VLAN interface. • Enter the VLAN number.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	Displays snooping configuration.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your configuration to the startup configuration.

Statically Configuring an Interface to Join a Group

Ports normally join multicast groups through the IGMP report message, but you can also statically configure a host on an interface.

Follow the steps below to add a port as a member of a multicast group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id*
4. **end**
5. **show mac-address-table multicast** [**vlan** *vlan-id*] [**user** | **igmp-snooping**] [**count**]
6. **show ip igmp snooping**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id* **Example:** `Device(config)# ip igmp snooping vlan 1 static 0100.5e05.0505 interface FastEthernet0/1/1`	Enables IGMP snooping on the VLAN interface.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show mac-address-table multicast** [**vlan** *vlan-id*] [**user** | **igmp-snooping**] [**count**] **Example:** `Device# show mac-address-table multicast vlan 1 igmp-snooping`	Displays MAC address table entries for a VLAN. • *vlan-id* is the multicast group VLAN ID. • **user** displays only the user-configured multicast entries. • **igmp-snooping** displays entries learned via IGMP snooping. • **count** displays only the total number of entries for the selected criteria, not the actual entries.
Step 6	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	Displays snooping configuration.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your configuration to the startup configuration.

Configuring a Multicast Device Port

Perform this task to enable a static connection to a multicast device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip igmp snooping vlan** *vlan-id* **mrouter** {**interface** *interface-id* | **learn pim-dvmrp**}
4. **end**
5. **show ip igmp snooping**
6. **show ip igmp snooping mrouter** [**vlan** *vlan-id*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **mrouter** {**interface** *interface-id* | **learn pim-dvmrp**} **Example:** `Device(config)# ip igmp snooping vlan1 interface Fa0/1/1 learn pim-dvmrp`	Enables IGMP snooping on the VLAN interface and enables route discovery.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	(Optional) Displays snooping configuration.
Step 6	**show ip igmp snooping mrouter** [**vlan** *vlan-id*] **Example:** `Device# show ip igmp snooping mroute vlan vlan1`	(Optional) Displays Mroute discovery information.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your configuration to the startup configuration.

Configuring Per-Port Storm Control

You can use these techniques to block the forwarding of unnecessary flooded traffic.

By default, unicast, broadcast, and multicast suppression is disabled.

Enabling Per-Port Storm Control

Perform this task to enable a per-port storm control.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **storm-control** {**broadcast** | **multicast** | **unicast**} **level** *level*
5. **storm-control action shutdown**
6. **storm-control action trap**
7. **end**
8. **show interfaces** *interface-type interface-number* **counters storm-control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/3/1`	Specifies the port to configure, and enters interface configuration mode. - Enter the interface type and interface number.
Step 4	**storm-control** {**broadcast** | **multicast** | **unicast**} **level** *level* **Example:** `Device(config-if)# storm-control broadcast level 7`	Configures broadcast, multicast, or unicast per-port storm control. - Specify the rising suppression level for either broadcast, multicast, or unicast traffic. The storm control action occurs when traffic utilization reaches this level.
Step 5	**storm-control action shutdown** **Example:** `Device(config-if)# storm-control action shutdown`	Selects the **shutdown** keyword to disable the port during a storm. - The default is to filter out the traffic.
Step 6	**storm-control action trap** **Example:** `Device(config-if)# storm-control action trap`	Sends Simple Management Network Protocol (SNMP) trap to disable the port during a storm. - The default is to filter out the traffic.

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 8	**show interfaces** *interface-type interface-number* **counters storm-control**	(Optional) Verifies your entries.

	Command or Action	Purpose
	Example: `Device# show interfaces fastethernet 0/3/1 counters storm-control`	

What to Do Next

Note If any type of traffic exceeds the upper threshold limit, all other traffic will be stopped.

Disabling Per-Port Storm Control

Perform this task to disable a per-port storm control.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **no storm-control** {**broadcast** | **multicast**| **unicast**} **level** *level*
5. **no storm-control action shutdown**
6. **no storm-control action trap**
7. **end**
8. **show interfaces** *interface-type interface-number* **counters storm-control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/3/1`	Specifies the interface and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**no storm-control** {**broadcast** | **multicast**| **unicast**} **level** *level* **Example:** `Device(config-if)# no storm-control broadcast level 7`	Disables per-port storm control.
Step 5	**no storm-control action shutdown** **Example:** `Device(config-if)# no storm-control action shutdown`	Disables the specified storm control action.
Step 6	**no storm-control action trap** **Example:** `Device(config-if)# no storm-control action trap`	Disables the specified storm control action.

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 8	**show interfaces** *interface-type interface-number* **counters storm-control** **Example:** `Device# show interfaces fastethernet 0/3/1 counters storm-control`	(Optional) Verifies your entries.

Configuring Stacking

Stacking is the connection of two device modules resident in the same chassis so that they behave as a single device. When a chassis is populated with two device modules, the user must configure to operate in stacked mode. This is done by selecting one port from each device module and configuring it to be a stacking partner. The user must then use a cable to connect the stacking partners from each device module to physically stack the device modules. Any one port in a device module can be designated as the stacking partner for that device module.

Perform this task to configure a pair of ports on two different device modules as stacking partners.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *interface-id*
4. **no shutdown**
5. **switchport stacking-partner interface fastethernet** *partner-interface-id*
6. **exit**
7. **interface** fastethernet *partner-interface-id*
8. **no shutdown**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *interface-id* **Example:** `Device(config)# interface fastethernet 0/3/1`	Enters interface configuration mode. • Enter the interface number.
Step 4	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Activates the interface. • This step is required only if you shut down the interface.
Step 5	**switchport stacking-partner interface fastethernet** *partner-interface-id* **Example:** `Device(config-if)# switchport stacking-partner interface FastEthernet partner-interface-id`	Selects and configures the stacking partner port. • Enter the partner interface number. • To restore the defaults, use the **no** form of this command.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to privileged configuration mode.
Step 7	**interface** fastethernet *partner-interface-id* **Example:** `Device# interface fastethernet 0/3/1`	Specifies the partner-interface, and enters interface configuration mode. - Enter the partner interface number.
Step 8	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Activates the stacking partner interface.

Step 9	**end**
	Example:
	`Device(config-if)# end`
		Exits configuration mode.

What to Do Next

Note Both stacking partner ports must have their **speed** and **duplex** parameters set to **auto**.

Caution If stacking is removed, stacked interfaces will shutdown. Other nonstacked ports will be left unchanged.

Configuring Fallback Bridging

The table below shows the default fallback bridging configuration.

Table 3: Default Fallback Bridging Configuration

Feature	Default Setting
Bridge groups	None are defined or assigned to an interface. No VLAN-bridge STP is defined.

Feature	Default Setting
Device forwards frames for stations that it has dynamically learned	Enabled.
Bridge table aging time for dynamic entries	300 seconds.
MAC-layer frame filtering	Disabled.
Spanning tree parameters: • Device priority • Interface priority • Interface path cost • Hello BPDU interval • Forward-delay interval • Maximum idle interval	• 32768 • 128 • 10 Mbps: 100 100 Mbps: 19 1000 Mbps: 4 • 2 seconds • 20 seconds • 30 seconds

Creating a Bridge Group

To configure fallback bridging for a set of switched virtual interfaces (SVIs), these interfaces must be assigned to bridge groups. All interfaces in the same group belong to the same bridge domain. Each SVI can be assigned to only one bridge group.

Perform this task to create a bridge group and assign an interface to it.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip routing**
4. **bridge** *bridge-group* **protocol vlan-bridge**
5. **interface** *interface-type interface-number*
6. **bridge-group** *bridge-group*
7. **end**
8. **show vlan-bridge**
9. **show running-config**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip routing** **Example:** `Device(config)# no ip routing`	Disables IP routing.
Step 4	**bridge** *bridge-group* **protocol vlan-bridge** **Example:** `Device(config)# bridge 100 protocol vlan-bridge`	Assigns a bridge group number and specifies the VLAN-bridge spanning-tree protocol to run in the bridge group. • The **ibm** and **dec** keywords are not supported. • For *bridge-group*, specify the bridge group number. The range is from 1 to 255. • Frames are bridged only among interfaces in the same group.
Step 5	**interface** *interface-type interface-number* **Example:** `Device(config)# interface vlan 0/3/1`	Specifies the interface on which you want to assign the bridge group, and enters interface configuration mode. • The specified interface must be an SVI: a VLAN interface that you created by using the **interface vlan** *vlan-id* global configuration command. • These ports must have IP addresses assigned to them.
Step 6	**bridge-group** *bridge-group* **Example:** `Device(config-if)# bridge-group 100`	Assigns the interface to the bridge group. • By default, the interface is not assigned to any bridge group. An interface can be assigned to only one bridge group.

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show vlan-bridge** **Example:** `Device# show vlan-bridge`	(Optional) Verifies forwarding mode.
Step 9	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entries.
Step 10	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Preventing the Forwarding of Dynamically Learned Stations

By default, the device forwards any frames for stations that it has dynamically learned. When this activity is disabled, the device only forwards frames whose addresses have been statically configured into the forwarding cache.

Perform this task to prevent the device from forwarding frames for stations that it has dynamically learned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no bridge** *bridge-group* **acquire**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no bridge** *bridge-group* **acquire** **Example:** **Example:** `Device(config)# no bridge 100 acquire`	Enables the device to stop forwarding any frames for stations that it has dynamically learned through the discovery process and to limit frame forwarding to statically configured stations. • The device filters all frames except those whose destined-to addresses have been statically configured into the forwarding cache. To configure a static address, use the **bridge** *bridge-group* **address** *mac-address* {**forward** | **discard**} global configuration command. • For *bridge-group*, specify the bridge group number. The range is 1 to 255.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Configuring the Bridge Table Aging Time

A device forwards, floods, or drops packets based on the bridge table. The bridge table maintains both static and dynamic entries. Static entries are entered by the user. Dynamic entries are entered by the bridge learning process. A dynamic entry is automatically removed after a specified length of time, known as aging time, from the time the entry was created or last updated.

If you are likely to move hosts on a switched network, decrease the aging time to enable the device to quickly adapt to the change. If hosts on a switched network do not continuously send packets, increase the aging time to keep the dynamic entries for a longer time and thus reduce the possibility of flooding when the hosts send again.

Perform this task to configure the aging time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **aging-time** *seconds*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **aging-time** *seconds* **Example:** `Device(config)# bridge 100 aging-time 10000`	Specifies the length of time that a dynamic entry remains in the bridge table from the time the entry was created or last updated. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 0 to 1000000. The default is 300 seconds.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Filtering Frames by a Specific MAC Address

A device examines frames and sends them through the internetwork according to the destination address; a device does not forward a frame back to its originating network segment. You can use the software to configure specific administrative filters that filter frames based on information other than the paths to their destinations.

You can filter frames with a particular MAC-layer station destination address. Any number of addresses can be configured in the system without a performance penalty.

Perform this task to filter by the MAC-layer address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **address** *mac-address* {**forward** | **discard**} [*interface-id*]
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **address** *mac-address* {**forward** | **discard**} [*interface-id*]	Filters frames with a particular MAC-layer station source or destination address.

	Command or Action	Purpose
	Example: **Example:** `Device(config)# bridge 1 address 0800.cb00.45e9 forward ethernet 1`	• Enter the bridge-group number (the range is 1 to 255), the MAC address and the **forward** or **discard** keywords.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Adjusting Spanning-Tree Parameters

You might need to adjust certain spanning-tree parameters if the default values are not suitable for your device configuration. Parameters affecting the entire spanning tree are configured with variations of the **bridge** global configuration command. Interface-specific parameters are configured with variations of the **bridge-group** interface configuration command.

You can adjust spanning-tree parameters by performing any of the tasks in these sections:

- Changing the Device Priority, page 67
- Changing the Interface Priority, page 68
- Assigning a Path Cost, page 69
- Adjusting BPDU Intervals, page 71
- Adjusting the Interval Between Hello BPDUs, page 71
- Changing the Forward-Delay Interval, page 72
- Changing the Maximum-Idle Interval, page 73
- Disabling the Spanning Tree on an Interface, page 74

Note Only network administrators with a good understanding of how devices and STP function should make adjustments to spanning-tree parameters. Poorly planned adjustments can have a negative impact on performance.

Changing the Device Priority

You can globally configure the priority of an individual device when two devices tie for position as the root device, or you can configure the likelihood that a device will be selected as the root device. This priority is determined by default; however, you can change it.

Perform this task to change the device priority.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **priority** *number*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **priority** *number* **Example:** `Device(config)# bridge 100 priority 5`	Changes the priority of the device. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *number*, enter a number from 0 to 65535. The default is 32768. The lower the number, the more likely the device will be chosen as the root.

	Command or Action	Purpose

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entry.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Changing the Interface Priority

You can change the priority for an interface. When two devices tie for position as the root device, you configure an interface priority to break the tie. The device with the lower interface value is elected.

Perform this task to change the interface priority.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **bridge** *bridge-group* **priority** *number*
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/3/1`	Specifies the interface to set the priority, and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**bridge** *bridge-group* **priority** *number* **Example:** `Device(config-if)# bridge 100 priority 4`	Changes the priority of the bridge. • Enter the bridge-group number and the priority number.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Assigning a Path Cost

Each interface has a path cost associated with it. By convention, the path cost is 1000/data rate of the attached LAN, in Mbps.

Perform this task to assign a path cost.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **bridge** *bridge-group* **path-costs** *cost*
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/3/1`	Specifies the interface to set the priority and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**bridge** *bridge-group* **path-costs** *cost* **Example:** `Device(config-if)# bridge 100 pathcost 4`	Changes the path cost. • Enter the bridge-group number and cost.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.

	Command or Action	Purpose
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Adjusting BPDU Intervals

You can adjust bridge protocol data unit (BPDU) intervals as described in these sections:

- Adjusting the Interval Between Hello BPDUs, page 71 (optional)
- Changing the Forward-Delay Interval, page 72 (optional)
- Changing the Maximum-Idle Interval, page 73 (optional)

Note Each device in a spanning tree adopts the interval between hello BPDUs, the forward delay interval, and the maximum idle interval parameters of the root device, regardless of what its individual configuration might be.

Adjusting the Interval Between Hello BPDUs

Perform this task to adjust the interval between hello BPDUs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **hello-time** *seconds*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **hello-time** *seconds* **Example:** `Device(config)# bridge 100 hello-time 5`	Specifies the interval between hello BPDUs. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 1 to 10. The default is 2 seconds.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Changing the Forward-Delay Interval

The forward-delay interval is the amount of time spent listening for topology change information after an interface has been activated for switching and before forwarding actually begins.

Perform this task to change the forward-delay interval.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **forward-time** *seconds*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **forward-time** *seconds* **Example:** `Device(config)# bridge 100 forward-time 25`	Specifies the forward-delay interval. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 10 to 200. The default is 20 seconds.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Changing the Maximum-Idle Interval

If a device does not hear BPDUs from the root device within a specified interval, it recomputes the spanning-tree topology.

Perform this task to change the maximum-idle interval (maximum aging time).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge** *bridge-group* **max-age** *seconds*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bridge** *bridge-group* **max-age** *seconds* **Example:** `Device(config)# bridge 100 forward-time 25`	Specifies the interval the device waits to hear BPDUs from the root device. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 10 to 200. The default is 30 seconds.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Disabling the Spanning Tree on an Interface

When a loop-free path exists between any two switched subnetworks, you can prevent BPDUs generated in one switching subnetwork from impacting devices in the other switching subnetwork, yet still permit switching throughout the network as a whole. For example, when switched LAN subnetworks are separated by a WAN, BPDUs can be prevented from traveling across the WAN link.

Perform this task to disable spanning tree on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **bridge-group** *bridge-group* **spanning-disabled**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/3/1`	Specifies the interface to set the priority and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	**bridge-group** *bridge-group* **spanning-disabled** **Example:** `Device(config-if)# bridge 100 spanning-disabled`	Disables spanning tree on the interface. • For *bridge-group*, specify the bridge group number. The range is 1 to 255.

	Command or Action	Purpose

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entry.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entry in the configuration file.

Monitoring and Maintaining the Network

Perform this task to monitor and maintain the network.

SUMMARY STEPS

1. **enable**
2. **clear bridge** *bridge-group*
3. **show bridge**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bridge** *bridge-group* **Example:** `Device# clear bridge bridge1`	(Optional) Removes any learned entries from the forwarding database and clears the transmit and receive counts for any statically configured entries. • Enter the number of the bridge group.

	Command or Action	Purpose
Step 3	**show bridge** **Example:** `Device# show bridge`	(Optional) Displays classes of entries in the bridge forwarding database.

Step 4	**end**
	Example:
	`Device# end`
		(Optional) Exits privileged EXEC mode.

Configuring Separate Voice and Data Subnets

The HWICs can automatically configure voice VLANs. This capability overcomes the management complexity of overlaying a voice topology onto a data network while maintaining the quality of voice traffic. With the automatically configured voice VLAN feature, network administrators can segment phones into separate logical networks, even though the data and voice infrastructure is physically the same. The voice VLAN feature places the phones into their own VLANs without the need for end-user intervention. A user can plug the phone into the device, which provides with the necessary VLAN information.

For ease of network administration and increased scalability, network managers can configure the HWICs to support Cisco IP phones such that the voice and data traffic reside on separate subnets. You should always use separate VLANs when you are able to segment the existing IP address space of your branch office.

User priority bits in the 802.1p portion of the 802.1Q standard header are used to provide prioritization in Ethernet devices. This is a vital component in designing Cisco AVVID networks.

The HWICs provides the performance and intelligent services of Cisco software for branch office applications. The HWICs can identify user applications--such as voice or multicast video--and classify traffic with the appropriate priority levels.

Follow these steps to automatically configure Cisco IP phones to send voice traffic on the voice VLAN ID (VVID) on a per-port basis (see the “Voice Traffic and VVID” section).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **switchport mode trunk**
5. **switchport voice vlan** *vlan-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**interface** *interface-type interface-number*
	Example:
	`Device(config)#`
	`interface fastethernet 0/2/1`
		Specifies the port to be configured and enters interface configuration mode.
		• Enter the interface type and interface number.

Step 4	**switchport mode trunk**
	Example:
	`Device(config-if)#`
	`switchport mode trunk`
		Configures the port to trunk mode.

Step 5	**switchport voice vlan** *vlan-id*
	Example:
	`Device(config-if)#`
	`switchport voice vlan 100`
		Configures the voice port with a VVID that will be used exclusively for voice traffic.
		• Enter the VLAN number.

Configuring a Single Subnet for Voice and Data

For network designs with incremental IP telephony deployment, network managers can configure the HWICs so that the voice and data traffic coexist on the same subnet. This might be necessary when it is impractical either to allocate an additional IP subnet for IP phones or to divide the existing IP address space into an additional subnet at the remote branch, it might be necessary to use a single IP address space for branch offices. (This is one of the simpler ways to deploy IP telephony.)

This configuration approach must address two key considerations:

- Network managers should ensure that existing subnets have enough available IP addresses for the new Cisco IP phones, each of which requires a unique IP address.
- Administering a network with a mix of IP phones and workstations on the same subnet might pose a challenge.

Perform this task to automatically configure Cisco IP phones to send voice and data traffic on the same VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. switchport access vlan vlan-id
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface fastethernet 0/2/1`	Specifies the port to be configured, and enters interface configuration mode. • Enter the interface type and interface number.
Step 4	switchport access vlan vlan-id **Example:** `Device(config-if)# switchport access vlan 100`	Sets the native VLAN for untagged traffic. • The value of *vlan-id* represents the ID of the VLAN that is sending and receiving untagged traffic on the port. Valid IDs are from 1 to 1001. Leading zeroes are not permitted.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Managing the EtherSwitch HWIC

Adding Trap Managers

A trap manager is a management station that receives and processes traps. When you configure a trap manager, community strings for each member device must be unique. If a member device has an IP address assigned to it, the management station accesses the device by using its assigned IP address.

By default, no trap manager is defined, and no traps are issued.

Perform this task to add a trap manager and community string.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** *ip-address traps* **snmp** *vlan-membership*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**snmp-server host** *ip-address traps* **snmp** *vlan-membership*
	Example:
	`Device(config)#`
	`snmp-server host 172.16.128.263 traps1 snmp`
	`vlancommunity1`
		Enters the trap manager IP address, community string, and the traps to generate.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configuring IP Information

This section describes how to assign IP information on the HWICs. The following topics are included:

Assigning IP Information to the Device

You can use a BOOTP server to automatically assign IP information to the device; however, the BOOTP server must be set up in advance with a database of physical MAC addresses and corresponding IP addresses, subnet masks, and default gateway addresses. In addition, the device must be able to access the BOOTP server through one of its ports. At startup, a device without an IP address requests the information from the BOOTP server; the requested information is saved in the device running the configuration file. To ensure that the IP information is saved when the device is restarted, save the configuration by entering the **write memory** command in privileged EXEC mode.

You can change the information in these fields. The mask identifies the bits that denote the network number in the IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. The broadcast address is reserved for sending messages to all hosts. The CPU sends traffic to an unknown IP address through the default gateway.

Perform this task to enter the IP information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. ip address ip-address subnet-mask
5. **exit**
6. ip default-gateway ip-address
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface vlan 1`	Specifies the interface (in this case, the VLAN) to which the IP information is assigned and enters interface configuration mode. • Enter the interface type and interface number. • VLAN 1 is the management VLAN, but you can configure any VLAN from IDs 1 to 1001.
Step 4	ip address ip-address subnet-mask **Example:** `Device(config-if)# ip address 192.168.2.10 255.255.255.255`	Specifies the IP address. • Enter the IP address and subnet mask.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 6	ip default-gateway ip-address **Example:** `Device(config)# ip default-gateway 192.168.2.20`	Sets the IP address of the default device. • Enter the IP address of the default device.

Step 7	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Removing IP Information From a Device

Use the following procedure to remove the IP information (such as an IP address) from a device.

Note Using the **no ip address** command in interface configuration mode disables the IP protocol stack and removes the IP information. Cluster members without IP addresses rely on the IP protocol stack being enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **no ip address**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Device(config)# interface vlan 1`	Specifies the interface (in this case, the VLAN) to which the IP information is assigned and enters interface configuration mode. • Enter the interface type and interface number. • VLAN 1 is the management VLAN, but you can configure any VLAN from IDs 1 to 1001.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes the IP address and subnet mask.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

What to Do Next

Danger If you are removing the IP address through a telnet session, your connection to the device will be lost .

Specifying a Domain Name and Configuring the DNS

Each unique IP address can have a host name associated with it. The Cisco software maintains an EXEC mode and related Telnet support operations. This cache speeds the process of converting names to addresses.

IP defines a hierarchical naming scheme that allows a device to be identified by its location or domain. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco Systems is a commercial organization that IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the FTP system, for example, is identified as *ftp.cisco.com*.

To track domain names, IP has defined the concept of a domain name server (DNS), the purpose of which is to hold a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, you must first identify the host names and then specify a name server and enable the DNS, the Internet's global naming scheme that uniquely identifies network devices.

Specifying the Domain Name

You can specify a default domain name that the software uses to complete domain name requests. You can specify either a single domain name or a list of domain names. When you specify a domain name, any IP host name without a domain name has that domain name appended to it before being added to the host table.

Specifying a Name Server

You can specify up to six hosts that can function as a name server to supply name information for the DNS.

Enabling the DNS

If your network devices require connectivity with devices in networks for which you do not control name assignment, you can assign device names that uniquely identify your devices within the entire internetwork. The Internet's global naming scheme, the DNS, accomplishes this task. This service is enabled by default.

Enabling Switched Port Analyzer

You can monitor traffic on a given port by forwarding incoming and outgoing traffic on the port to another port in the same VLAN. A Switched Port Analyzer (SPAN) cannot monitor ports in a different VLAN, and a SPAN port must be a static-access port. Any number of ports can be defined as SPAN ports, and any combination of ports can be monitored. SPAN is supported for up to 2 sessions.

Perform this task to enable SPAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **monitor session** session-id {**destination** | **source**} {**interface** | **vlan** *interface-id* | *vlan-id*}} [**,** | **-** | **both** | **tx** | **rx**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**monitor session** session-id {**destination** | **source**} {**interface** | **vlan** *interface-id* | *vlan-id*}} [**,** | **-** | **both** | **tx** | **rx**]
	Example:
	`Device(config)#`
	`monitor session session-id {destination | source}`
	`{interface | vlan interface-id | vlan-id}} [, | -`
	`| both | tx | rx]`
		Enables port monitoring for a specific session (“*number*”).
		- Optionally, supply a SPAN *destination* interface and a *source* interface.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Disabling SPAN

Perform this task to disable SPAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. no monitor session session-id
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	no monitor session session-id
	Example:
	`Device(config)# no monitor session`
	`37`
		Disables port monitoring for a specific session.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Managing the ARP Table

To communicate with a device (on Ethernet, for example), the software first must determine the 48-bit MAC or local data link address of that device. The process of determining the local data link address from an IP address is called *address resolution*.

The Address Resolution Protocol (ARP) associates a host IP address with the corresponding media or MAC addresses and VLAN ID. Taking an IP address as input, ARP determines the associated MAC address. Once a MAC address is determined, the IP-MAC address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP). By default, standard Ethernet-style ARP encapsulation (represented by the **arpa** keyword) is enabled on the IP interface.

When you manually add entries to the ARP table by using the CLI, you must be aware that these entries do not age and must be manually removed.

Managing the MAC Address Tables

This section describes how to manage the MAC address tables on the HWICs. The following topics are included:

- Understanding MAC Addresses and VLANs

- Changing the Address Aging Time
- Configuring the Aging Time

The device uses the MAC address tables to forward traffic between ports. All MAC addresses in the address tables are associated with one or more ports. These MAC tables include the following types of addresses:

- Dynamic address--A source MAC address that the device learns and then drops when it is not in use.
- Secure address--A manually entered unicast address that is usually associated with a secured port. Secure addresses do not age.
- Static address--A manually entered unicast or multicast address that does not age and that is not lost when the device resets.

The address tables list the destination MAC address and the associated VLAN ID, module, and port number associated with the address. The following shows an example of a list of addresses as they would appear in the dynamic, secure, or static address table.

```
Device# show mac-address-table
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
000a.000b.000c          Secure      1      FastEthernet0/1/8
000d.e105.cc70          Self        1      Vlan1
00aa.00bb.00cc          Static      1      FastEthernet0/1/0
```

All addresses are associated with a VLAN. An address can exist in more than one VLAN and have different destinations in each. Multicast addresses, for example, could be forwarded to port 1 in VLAN 1 and ports 9, 10, and 11 in VLAN 5.

Each VLAN maintains its own logical address table. A known address in one VLAN is unknown in another until it is learned or statically associated with a port in the other VLAN. An address can be secure in one VLAN and dynamic in another. Addresses that are statically entered in one VLAN must be static addresses in all other VLANs.

Dynamic addresses are source MAC addresses that the device learns and then drops when they are not in use. Use the Aging Time field to define how long the device retains unseen addresses in the table. This parameter applies to all VLANs.

Setting too short an aging time can cause addresses to be prematurely removed from the table. Then when the device receives a packet for an unknown destination, it floods the packet to all ports in the same VLAN as the receiving port. This unnecessary flooding can impact performance. Setting too long an aging time can cause the address table to be filled with unused addresses; it can cause delays in establishing connectivity when a workstation is moved to a new port.

Perform this task to configure the dynamic address table aging time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. mac-address-table aging-time seconds
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	mac-address-table aging-time seconds **Example:** `Device(config)# mac-address-table aging-time 30000`	Enters the number of seconds that dynamic addresses are to be retained in the address table. • Valid entries are from 10 to 1000000.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Removing Dynamic Addresses

Follow these steps to remove a dynamic address entry.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. no mac-address-table dynamic hw-addr
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	no mac-address-table dynamic hw-addr **Example:** `Device(config)# no mac-address-table dynamic 0100.5e05.0505`	Enters the MAC address to be removed from dynamic MAC address table.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Adding Secure Addresses

The secure address table contains secure MAC addresses and their associated ports and VLANs. A secure address is a manually entered unicast address that is forwarded to only one port per VLAN. If you enter an address that is already assigned to another port, the device reassigns the secure address to the new port.

You can enter a secure port address even when the port does not yet belong to a VLAN. When the port is later assigned to a VLAN, packets destined for that address are forwarded to the port.

Note When you change the VLAN ID for a port that is configured with a secure MAC address, you must reconfigure the secure MAC address to reflect the new VLAN association.

Perform this task to add a secure address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. mac-address-table secure **address** hw-addr **interface** *interface-id*vlan vlan-id
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	mac-address-table secure **address** hw-addr **interface** *interface-id*vlan vlan-id
	Example:
	`Device(config)#`
	`mac-address-table secure address 0100.5e05.0505`
	`interface 0/3/1 vlan vlan 1`
		Enters the MAC address, its associated port, and the VLAN ID.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Removing a Secure Address

Perform this task to remove a secure address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. no mac-address-table secure hw-addr vlan vlan-id
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	no mac-address-table secure hw-addr vlan vlan-id **Example:** `Device(config)# no mac-address-table secure address 0100.5e05.0505 vlan vlan 1`	Enters the secure MAC address, its associated port, and the VLAN ID to be removed.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configuring Static Addresses

A static address has the following characteristics:

- It is manually entered in the address table and must be manually removed.
- It can be a unicast or multicast address.
- It does not age and is retained when the device restarts.

Because all ports are associated with at least one VLAN, the device acquires the VLAN ID for the address from the ports that you select on the forwarding map. A static address in one VLAN must be a static address in other VLANs. A packet with a static address that arrives on a VLAN where it has not been statically entered is flooded to all ports and not learned.

Perform this task to add a static address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. mac-address-table static hw-addr [interface] interface-id [vlan] vlan-id
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	mac-address-table static hw-addr [interface] interface-id [vlan] vlan-id
	Example:
	`Device(config)#`
	`mac-address-table static 0100.5e05.0505 interface`
	` 0/3/1 vlan vlan 1`
		Enters the static MAC address, the interface, and the VLAN ID of those ports.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Removing a Static Address

Follow these steps to remove a static address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. no mac-address-table static hw-addr [interface] interface-id [vlan] vlan-id
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	no mac-address-table static hw-addr [interface] interface-id [vlan] vlan-id
	Example:
	`Device(config)#`
	`no mac-address-table static 0100.5e05.0505`
	`interface 0/3/1 vlan vlan`
		Enters the static MAC address, the interface, and the VLAN ID of the port to be removed.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Clearing All MAC Address Tables

Perform this task to remove all MAC address tables.

SUMMARY STEPS

1. **enable**
2. **clear mac-address-table**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**clear mac-address-table** **Example:** `Device# clear mac-address-table`	Clears all MAC address tables.

Step 3	**end**
	Example:
	`Device# end`
		Exits privileged EXEC mode.

Configuration Examples for EtherSwitch HWICs

Range of Interface Examples

Example: Single Range Configuration

The following example shows all Fast Ethernet interfaces on an HWIC-4ESW in slot 2 being reenabled:

```
Device(config)# interface range fastethernet 0/3/0 - 8
Device(config-if-range)# no shutdown
Device(config-if-range)#
*Mar  21 14:01:21.474: %LINK-3-UPDOWN: Interface FastEthernet0/3/0, changed state to up
*Mar  21 14:01:21.490: %LINK-3-UPDOWN: Interface FastEthernet0/3/1, changed state to up
*Mar  21 14:01:21.502: %LINK-3-UPDOWN: Interface FastEthernet0/3/2, changed state to up
*Mar  21 14:01:21.518: %LINK-3-UPDOWN: Interface FastEthernet0/3/3, changed state to up
*Mar  21 14:01:21.534: %LINK-3-UPDOWN: Interface FastEthernet0/3/4, changed state to up
*Mar  21 14:01:21.546: %LINK-3-UPDOWN: Interface FastEthernet0/3/5, changed state to up
*Mar  21 14:01:21.562: %LINK-3-UPDOWN: Interface FastEthernet0/3/6, changed state to up
*Mar  21 14:01:21.574: %LINK-3-UPDOWN: Interface FastEthernet0/3/7, changed state to up
*Mar  21 14:01:21.590: %LINK-3-UPDOWN: Interface FastEthernet0/3/8, changed state to up
Device(config-if-range)#
```

Example: Range Macro Definition

The following example shows how to define an interface-range macro named enet_list to select Fast Ethernet interfaces 0/1/0 through 0/1/3:

```
Device(config)# define interface-range enet_list fastethernet 0/1/0 - 0/1/3
```

The following example shows how to define an interface-range configuration mode using the interface-range macro enet_list:

```
Device(config)# interface-range
  macro
```

```
enet_list
```

Optional Interface Feature Examples

Example: Interface Speed

The following example shows how to set the interface speed to 100 Mbps on Fast Ethernet interface 0/3/7:

```
Device(config)# interface fastethernet 0/3/7
Device(config-if)# speed 100
```

Example: Setting the Interface Duplex Mode

The following example shows how to set the interface duplex mode to full on Fast Ethernet interface 0/3/7:

```
Device(config)# interface fastethernet 0/3/7
Device(config-if)# duplex full
```

Example: Adding a Description for an Interface

The following example shows how to add a description of Fast Ethernet interface 0/3/7:

```
Device(config)# interface fastethernet 0/3/7
Device(config-if)# description Link to root device
```

Example: Stacking

The following example shows how to stack two HWICs.

```
Device(config)# interface FastEthernet 0/1/8
Device(config-if)# no shutdown
Device(config-if)# switchport stacking-partner interface FastEthernet 0/3/8
Device(config-if)# interface FastEthernet 0/3/8
Device(config-if)# no shutdown
```

Note In practice, the command **switchport stacking-partner interface FastEthernet** *0/partner-slot/partner-port* needs to be executed for only one of the stacked ports. The other port will be automatically configured as a stacking port by the Cisco software. The command **no shutdown**, however, must be executed for both of the stacked ports.

Example: VLAN Configuration

The following example shows how to configure inter-VLAN routing:

```
Device> enable
Device# configure terminal
Device(config)# vlan 45
Device(config-vlan)# vlan 1
Device(config-vlan)# vlan 2
Device(config-vlan)# exit
Device# configure terminal
Device(config)# interface vlan 1
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# no shut
Device(config-if)# interface vlan 2
Device(config-if)# ip address 10.2.2.2 255.255.255.0
Device(config-if)# no shut
Device(config-if)# interface FastEthernet 0/1/0
Device(config-if)# switchport access vlan 1
Device(config-if)# interface Fast Ethernet 0/1/1
Device(config-if)# switchport access vlan 2
Device(config-if)# exit
```

Example: VLAN Trunking Using VTP

The following example shows how to configure the device as a VTP server:

```
Device# vlan database
Device(vlan)# vtp server
Setting device to VTP SERVER mode.
Device(vlan)# vtp domain Lab
_Network
Setting VTP domain name to Lab_Network
Device(vlan)# vtp password WATER
Setting device VLAN database password to WATER.
Device(vlan)# exit
APPLY completed.
Exiting....
Device#
```

The following example shows how to configure the device as a VTP client:

```
Device# vlan database
Device(vlan)# vtp client
Setting device to VTP CLIENT mode.
Device(vlan)# exit
In CLIENT state, no apply attempted.
Exiting....
Device#
```

The following example shows how to configure the device as VTP transparent:

```
Device# vlan database
Device(vlan)# vtp transparent
Setting device to VTP TRANSPARENT mode.
Device(vlan)# exit
APPLY completed.
Exiting....
Device#
```

Spanning Tree Examples

Example: Configuring Spanning Tree Port Priority

The following example shows how to configure VLAN port priority on an interface:

```
Device# configure terminal
Device(config)# interface fastethernet 0/3/2
Device(config-if)# spanning-tree vlan 20 port priority 64
Device(config-if)# end
```

The following example shows how to verify the configuration of VLAN 20 on an interface when it is configured as a trunk port:

```
Device#show spanning-tree vlan 20

 VLAN20 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 00ff.ff90.3f54
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 32768, address 00ff.ff10.37b7
  Root port is 33 (FastEthernet0/3/2), cost of root path is 19
  Topology change flag not set, detected flag not set
  Number of topology flags 0 last change occurred 00:05:50 ago
  Times: hold 1, topology change 35, notification 2
     hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0, aging 0
 Port 33 (FastEthernet0/3/2) of VLAN20 is forwarding
  Port path cost 18, Port priority 64, Port Identifier 64.33
  Designated root has priority 32768, address 00ff.ff10.37b7
  Designated bridge has priority 32768, address 00ff.ff10.37b7
  Designated port id is 128.13, designated path cost 0
  Timers: message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  BPDU: sent 1, received 175
```

Example: Configuring Spanning Tree Port Cost

The following example shows how to change the spanning tree port cost of a Fast Ethernet interface:

```
Device# configure terminal
Device(config)# interface fastethernet0/3/2
Device(config-if)# spanning-tree cost 18
Device(config-if)# end
Device#
Device# show run interface fastethernet0/3/2
Building configuration...
Current configuration: 140 bytes
!
interface FastEthernet0/3/2
 switchport access vlan 20
  no ip address
  spanning-tree vlan 20 port-priority 64
  spanning-tree cost 18
end
```

The following example shows how to verify the configuration of a Fast Ethernet interface when it is configured as an access port:

```
Device# show spanning-tree interface fastethernet0/3/2

 Port 33 (FastEthernet0/3/2) of VLAN20 is forwarding
  Port path cost 18, Port priority 64, Port Identifier 64.33
  Designated root has priority 32768, address 00ff.ff10.37b7
  Designated bridge has priority 32768, address 00ff.ff10.37b7
  Designated port id is 128.13, designated path cost 0
  Timers: message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  BPDU: sent 1, received 175
```

Example: Configuring the Bridge Priority of a VLAN

The following example shows how to configure the bridge priority of VLAN 20 to 33792:

```
Device# configure terminal
Device(config)# spanning-tree vlan 20 priority 33792
Device(config)# end
```

Example: Configuring Hello Time

The following example shows how to configure the hello time for VLAN 20 to 7 seconds:

```
Device# configure terminal
Device(config)# spanning-tree vlan 20 hello-time 7
Device(config)# end
```

Example: Configuring the Forward Delay Time for a VLAN

The following example shows how to configure the forward delay time for VLAN 20 to 21 seconds:

```
Device#configure terminal
Device(config)#spanning-tree vlan 20 forward-time 21
Device(config)#end
```

Example: Configuring the Maximum Aging Time for a VLAN

The following example shows how to configure the maximum aging time for VLAN 20 to 36 seconds:

```
Device#configure terminal
Device(config)#spanning-tree vlan 20 max-age 36
Device(config)#end
```

Example: Enabling Spanning Tree Protocol

The following example shows how to enable spanning tree protocol on VLAN 20:

```
Device# configure terminal
Device(config)# spanning-tree vlan 20
Device(config)# end
Device#
```

Note Because spanning tree is enabled by default, the **show running** command will not display the command you entered to enable spanning tree protocol.

The following example shows how to disable spanning tree protocol on VLAN 20:

```
Device# configure terminal
Device(config)# no spanning-tree vlan 20
Device(config)# end
Device#
```

Example: Configuring Spanning Tree Root Bridge

The following example shows how to configure the spanning tree root bridge for VLAN 10, with a network diameter of 4:

```
Device# configure terminal
Device(config)# spanning-tree vlan 10 root primary diameter 4
Device(config)# exit
```

Example: MAC Table Manipulation

The following example shows how to configure a static entry in the MAC address table:

```
Device(config)# mac-address-table static beef.beef.beef interface fastethernet 0/1/5
Device(config)# end
```

The following example shows how to configure the port security in the MAC address table.

```
Device(config)# mac-address-table secure 0000.1111.2222 fastethernet 0/1/2 vlan 3
Device(config)# end
```

Switched Port Analyzer (SPAN) Source Examples

Example: SPAN Source Configuration

The following example shows how to configure the SPAN session 1 to monitor bidirectional traffic from source interface Fast Ethernet 0/1/1:

```
Device(config)# monitor session 1 source interface fastethernet 0/1/1
```

Example: SPAN Destination Configuration

The following example shows how to configure Fast Ethernet 0/3/7 interface as the destination for SPAN session 1:

```
Device(config)# monitor session 1 destination interface fastethernet 0/3/7
```

Example: Removing Sources or Destinations from a SPAN Session

This following example shows interface Fast Ethernet 0/3/2 being removed as a SPAN source for SPAN session 1:

```
Device(config)# no monitor session 1 source interface fastethernet 0/3/2
```

Example: IGMP Snooping

The following example shows the output from configuring IGMP snooping:

```
Device# show mac-address-table multicast igmp-snooping
```

```
HWIC Slot: 1
--------------
    MACADDR     VLANID     INTERFACES
0100.5e05.0505    1        Fa0/1/1
0100.5e06.0606    2
HWIC Slot: 3
--------------
    MACADDR     VLANID     INTERFACES
0100.5e05.0505    1        Fa0/3/4
0100.5e06.0606    2        Fa0/3/0
Device#
```

The following is an example of output from the **show running interface** privileged EXEC command for VLAN 1:

```
Device#
show running interface vlan 1
Building configuration...
Current configuration :82 bytes
!
interface Vlan1
 ip address 192.168.4.90 255.255.255.0
 ip pim sparse-mode
end
Device#
show running interface vlan 2

Building configuration...
Current configuration :82 bytes
!
interface Vlan2
 ip address 192.168.5.90 255.255.255.0
 ip pim sparse-mode
end
Device#
Device# show ip igmp group
IGMP Connected Group Membership
Group Address    Interface                 Uptime    Expires   Last Reporter
209.165.200.225 Vlan1                    01:06:40  00:02:20  192.168.41.101
209.165.200.226 Vlan2                    01:07:50  00:02:17  192.168.5.90
209.165.200.227 Vlan1                    01:06:37  00:02:25  192.168.41.100
209.165.200.228 Vlan2                    01:07:40  00:02:21  192.168.31.100
209.165.200.229 Vlan1                    01:06:36  00:02:22  192.168.41.101
209.165.200.230 Vlan2                    01:06:39  00:02:20  192.168.31.101
Device#
Device# show ip mroute
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C -
Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode
(*, 209.165.200.230), 01:06:43/00:02:17, RP 0.0.0.0, flags:DC
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse, 01:06:43/00:02:17
(*, 209.165.200.226), 01:12:42/00:00:00, RP 0.0.0.0, flags:DCL
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan2, Forward/Sparse, 01:07:53/00:02:14
(*, 209.165.200.227), 01:07:43/00:02:22, RP 0.0.0.0, flags:DC
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse, 01:06:40/00:02:22
    Vlan2, Forward/Sparse, 01:07:44/00:02:17
```

(*, 209.165.200.2282), 01:06:43/00:02:18, RP 0.0.0.0, flags:DC

```
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse, 01:06:40/00:02:18
    Vlan2, Forward/Sparse, 01:06:43/00:02:16
Device#
```

Example: Storm-Control

The following example shows how to enable bandwidth-based multicast suppression at 70 percent on Fast Ethernet interface 2:

```
Device> enable
Device# configure terminal
Device(config)# interface FastEthernet0/3/3
Device(config-if)# storm-control multicast threshold 70.0 30.0
Device(config-if)# end
Device# show interfaces FastEthernet0/3/3 counters storm-control
Interface  Filter State  Upper    Lower    Current
---------  ------------  -----    -----    -------
Fa0/1/0    inactive      100.00%  100.00%  N/A
Fa0/1/1    inactive      100.00%  100.00%  N/A
Fa0/1/2    inactive      100.00%  100.00%  N/A
Fa0/1/3    inactive      100.00%  100.00%  N/A
Fa0/3/0    inactive      100.00%  100.00%  N/A
Fa0/3/1    inactive      100.00%  100.00%  N/A
Fa0/3/2    inactive      100.00%  100.00%  N/A
Fa0/3/3    Forwarding     70.00%   30.00%  0.00%
Fa0/3/4    inactive      100.00%  100.00%  N/A
Fa0/3/5    inactive      100.00%  100.00%  N/A
Fa0/3/6    inactive      100.00%  100.00%  N/A
Fa0/3/7    inactive      100.00%  100.00%  N/A
Fa0/3/8    inactive      100.00%  100.00%  N/A
```

Ethernet Switching Examples

Example: Subnets for Voice and Data

The following example shows how to configure separate subnets for voice and data on the EtherSwitch HWIC:

```
interface FastEthernet0/1/1
 description DOT1Q port to IP Phone
 switchport native vlan 50
 switchport mode trunk
 switchport voice vlan 150
interface Vlan 150
description voice vlan
ip address
209.165.200.227
 255.255.255.0
ip helper-address
209.165.200.228
 (See Note below)
interface Vlan 50
description data vlan
ip address
209.165.200.220
 255.255.255.0
```

This configuration instructs the IP phone to generate a packet with an 802.1Q VLAN ID of 150 that has 802.1p value of 5 (default for voice bearer traffic).

Note In a centralized CallManager deployment model, the DHCP server might be located across the WAN link. If so, an **ip helper-address** command pointing to the DHCP server should be included on the voice VLAN interface for the IP phone. This is done to obtain its IP address as well as the address of the TFTP server required for its configuration.

Be aware that Cisco software supports a DHCP server function. If this function is used, the EtherSwitch HWIC serves as a local DHCP server and a helper address would not be required.

Example: Inter-VLAN Routing

Configuring inter-VLAN routing is identical to the configuration on an EtherSwitch HWIC with an MSFC. Configuring an interface for WAN routing is consistent with other Cisco software platforms.

The following example provides a sample configuration:

```
interface Vlan 160
 description voice vlan
 ip address 10.6.1.1 255.255.255.0
interface Vlan 60
 description data vlan
 ip address 10.60.1.1 255.255.255.0
interface Serial0/3/0
 ip address 172.3.1.2 255.255.255.0
```

Note Standard IGP routing protocols such as RIP, IGRP, EIGRP, and OSPF are supported on the EtherSwitch HWIC. Multicast routing is also supported for PIM dense mode, sparse mode and sparse-dense mode.

Example: Single Subnet Configuration

The EtherSwitch HWIC supports the use of an 802.1p-only option when configuring the voice VLAN. Using this option allows the IP phone to tag VoIP packets with a Cost of Service of 5 on the native VLAN, while all PC data traffic is sent untagged

The following example shows a single subnet configuration for the EtherSwitch HWIC:

```
Device# FastEthernet 0/1/2
description Port to IP Phone in single subnet
 switchport access vlan 40
```

The EtherSwitch HWIC instructs the IP phone to generate an 802.1Q frame with a null VLAN ID value but with an 802.1p value (default is COS of 5 for bearer traffic). The voice and data VLANs are both 40 in this example.

Example: Ethernet Ports on IP Phones with Multiple Ports

The following example illustrates the configuration for the IP phone:

```
interface FastEthernet0/x/x
```

```
 switchport voice vlan x
 switchport mode trunk
```

The following example illustrates the configuration for the PC:

```
interface FastEthernet0/x/y
 switchport mode access
 switchport access vlan y
```

Note Using a separate subnet, and possibly a separate IP address space, may not be an option for some small branch offices due to the IP routing configuration. If the IP routing can handle an additional subnet at the remote branch, you can use Cisco Network Registrar and secondary addressing.

Additional References for IEEE 802.1Q Tunneling

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
IP LAN switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS LAN Switching Services Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the Cisco HWIC-4ESW and the Cisco HWIC-D-9ESW EtherSwitch Cards

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 4: Feature Information for the 4-Port Cisco HWIC-4ESW and the 9-Port Cisco HWIC-D-9ESW EtherSwitch High Speed WAN Interface Cards

Feature Name	Releases	Feature Information
4-port Cisco HWIC-4ESW and the 9-port Cisco HWIC-D-9ESW EtherSwitch high speed WAN interface cards (HWICs) hardware feature	12.3(8)T4	The 4-port Cisco HWIC-4ESW and the 9-port Cisco HWIC-D-9ESW EtherSwitch high speed WAN interface cards (HWICs) hardware feature is supported on Cisco 1800 (modular), Cisco 2800, and Cisco 3800 series integrated services devices. Cisco EtherSwitch HWICs are 10/100BASE-T Layer 2 Ethernet devices with Layer 3 routing capability. (Layer 3 routing is forwarded to the host and is not actually performed at the device.) Traffic between different VLANs on a device is routed through the device platform. Any one port on a Cisco EtherSwitch HWIC may be configured as a stacking port to link to another Cisco EtherSwitch HWIC or EtherSwitch network module in the same system. An optional power module can also be added to provide inline power for IP telephones. The HWIC-D-9ESW HWIC requires a double-wide card slot.

CHAPTER 3

Configuring Routing Between VLANs

This module provides an overview of VLANs. It describes the encapsulation protocols used for routing between VLANs and provides some basic information about designing VLANs. This module contains tasks for configuring routing between VLANS.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Routing Between VLANs

Virtual Local Area Network Definition

A virtual local area network (VLAN) is a switched network that is logically segmented on an organizational basis, by functions, project teams, or applications rather than on a physical or geographical basis. For example, all workstations and servers used by a particular workgroup team can be connected to the same VLAN, regardless of their physical connections to the network or the fact that they might be intermingled with other teams. Reconfiguration of the network can be done through software rather than by physically unplugging and moving devices or wires.

A VLAN can be thought of as a broadcast domain that exists within a defined set of switches. A VLAN consists of a number of end systems, either hosts or network equipment (such as bridges and routers), connected by a single bridging domain. The bridging domain is supported on various pieces of network equipment; for example, LAN switches that operate bridging protocols between them with a separate bridge group for each VLAN.

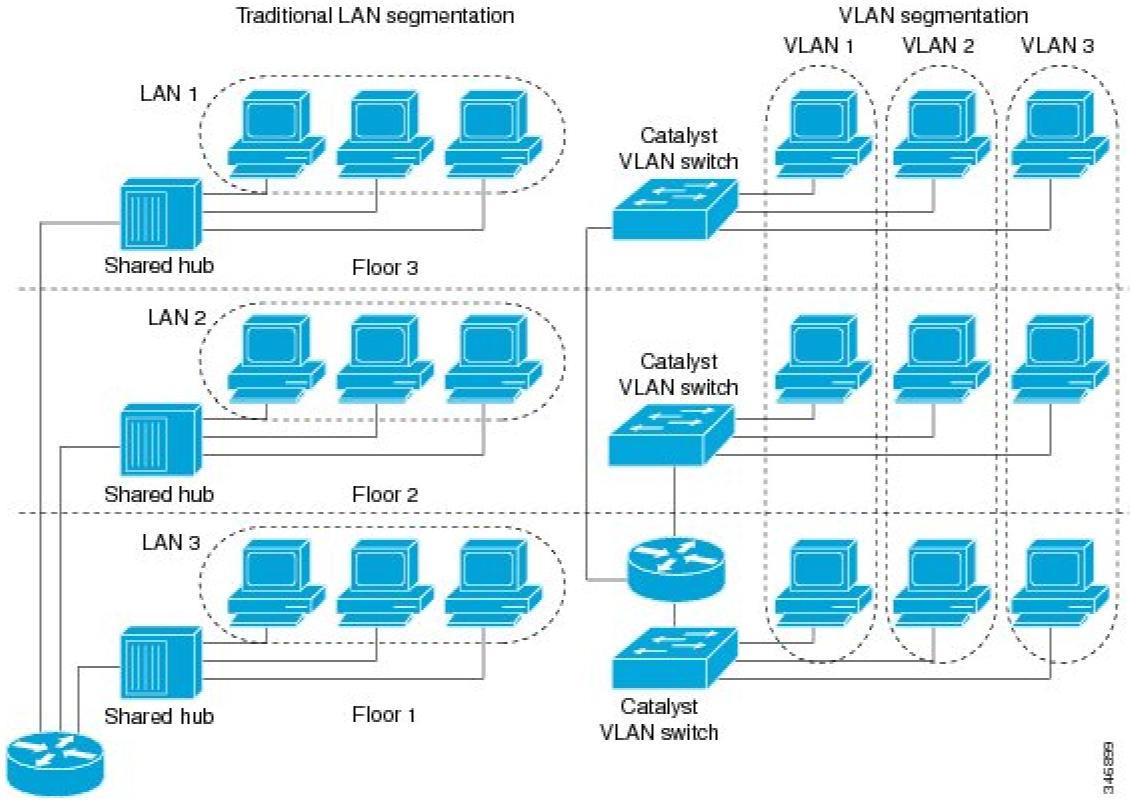
VLANs are created to provide the segmentation services traditionally provided by routers in LAN configurations. VLANs address scalability, security, and network management. Routers in VLAN topologies provide broadcast filtering, security, address summarization, and traffic flow management. None of the switches within the defined group will bridge any frames, not even broadcast frames, between two VLANs. Several key issues described in the following sections need to be considered when designing and building switched LAN internetworks:

LAN Segmentation

VLANs allow logical network topologies to overlay the physical switched infrastructure such that any arbitrary collection of LAN ports can be combined into an autonomous user group or community of interest. The technology logically segments the network into separate Layer 2 broadcast domains whereby packets are switched between ports designated to be within the same VLAN. By containing traffic originating on a particular LAN only to other LANs in the same VLAN, switched virtual networks avoid wasting bandwidth, a drawback inherent to traditional bridged and switched networks in which packets are often forwarded to LANs with no need for them. Implementation of VLANs also improves scalability, particularly in LAN environments that support broadcast- or multicast-intensive protocols and applications that flood packets throughout the network.

The figure below illustrates the difference between traditional physical LAN segmentation and logical VLAN segmentation.

Figure 1: LAN Segmentation and VLAN Segmentation



Security

VLANs improve security by isolating groups. High-security users can be grouped into a VLAN, possibly on the same physical segment, and no users outside that VLAN can communicate with them.

Broadcast Control

Just as switches isolate collision domains for attached hosts and only forward appropriate traffic out a particular port, VLANs provide complete isolation between VLANs. A VLAN is a bridging domain, and all broadcast and multicast traffic is contained within it.

VLAN Performance

The logical grouping of users allows an accounting group to make intensive use of a networked accounting system assigned to a VLAN that contains just that accounting group and its servers. That group’s work will not affect other users. The VLAN configuration improves general network performance by not slowing down other users sharing the network.

Network Management

The logical grouping of users allows easier network management. It is not necessary to pull cables to move a user from one network to another. Adds, moves, and changes are achieved by configuring a port into the appropriate VLAN.

Network Monitoring Using SNMP

SNMP support has been added to provide mib-2 interfaces sparse table support for Fast Ethernet subinterfaces. Monitor your VLAN subinterface using the **show vlans** EXEC command. For more information on configuring SNMP on your Cisco network device or enabling an SNMP agent for remote access, see the "Configuring SNMP Support" module in the *Cisco IOS Network Management Configuration Guide* .

Communication Between VLANs

Communication between VLANs is accomplished through routing, and the traditional security and filtering functions of the router can be used. Cisco IOS software provides network services such as security filtering, quality of service (QoS), and accounting on a per-VLAN basis. As switched networks evolve to distributed VLANs, Cisco IOS software provides key inter-VLAN communications and allows the network to scale.

Before Cisco IOS Release 12.2, Cisco IOS support for interfaces that have 802.1Q encapsulation configured is IP, IP multicast, and IPX routing between respective VLANs represented as subinterfaces on a link. New functionality has been added in IEEE 802.1Q support for bridging on those interfaces and the capability to configure and use integrated routing and bridging (IRB).

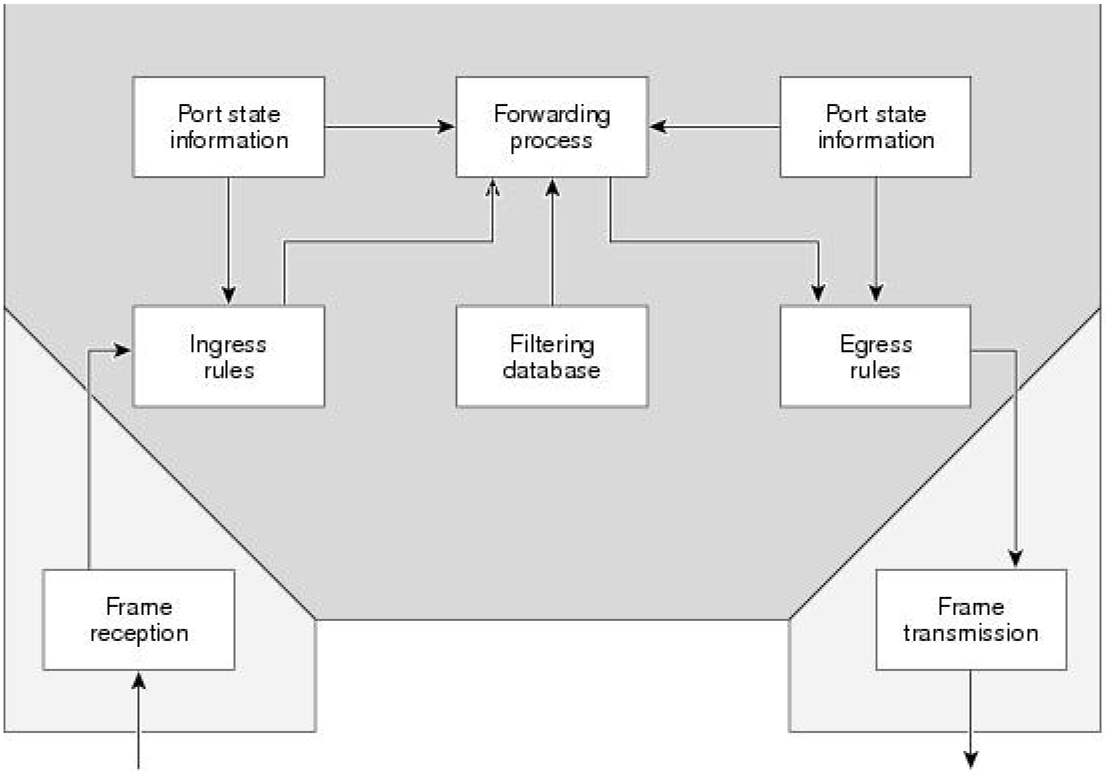
Relaying Function

The relaying function level, as displayed in the figure below, is the lowest level in the architectural model described in the IEEE 802.1Q standard and presents three types of rules:

- Ingress rules--Rules relevant to the classification of received frames belonging to a VLAN.
- Forwarding rules between ports--Rules decide whether to filter or forward the frame.

- Egress rules (output of frames from the switch)--Rules decide if the frame must be sent tagged or untagged.

Figure 2: Relaying Function

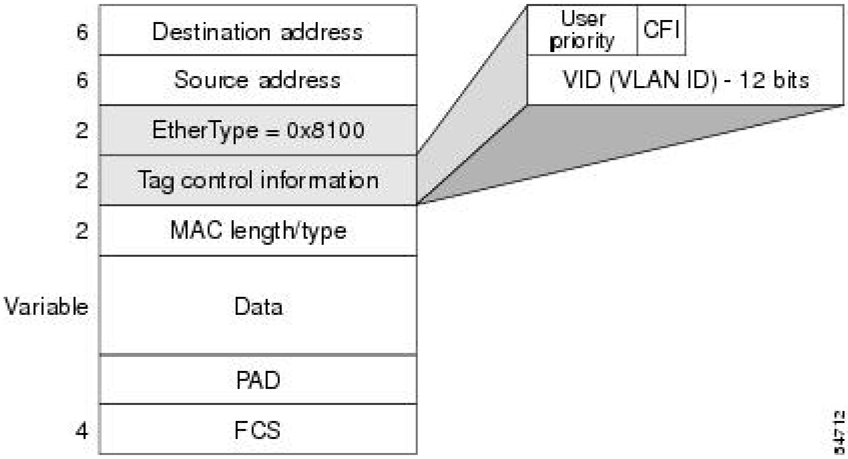


The Tagging Scheme

The figure below shows the tagging scheme proposed by the 802.3ac standard, that is, the addition of the four octets after the source MAC address. Their presence is indicated by a particular value of the EtherType field (called TPID), which has been fixed to be equal to 0x8100. When a frame has the EtherType equal to 0x8100, this frame carries the tag IEEE 802.1Q/802.1p. The tag is stored in the following two octets and it contains 3 bits of user priority, 1 bit of Canonical Format Identifier (CFI), and 12 bits of VLAN ID (VID). The 3 bits of user priority are used by the 802.1p standard; the CFI is used for compatibility reasons between Ethernet-type networks and Token Ring-type networks. The VID is the identification of the VLAN, which is basically used by the 802.1Q standard; being on 12 bits, it allows the identification of 4096 VLANs.

After the two octets of TPID and the two octets of the Tag Control Information field there are two octets that originally would have been located after the Source Address field where there is the TPID. They contain either the MAC length in the case of IEEE 802.3 or the EtherType in the case of Ethernet version 2.

Figure 3: Tagging Scheme

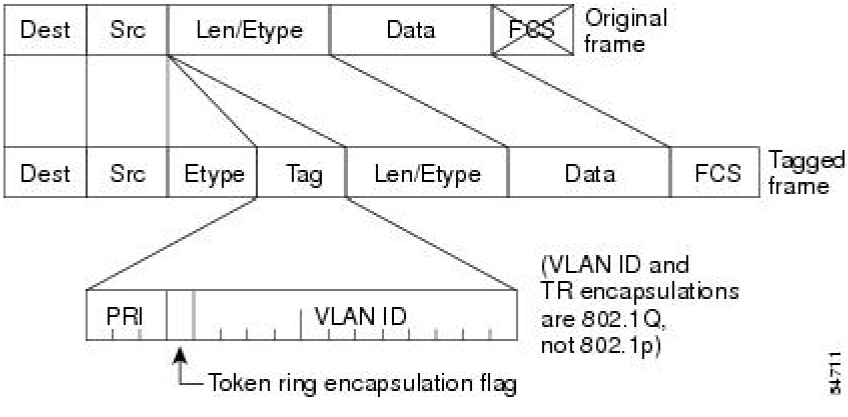


The EtherType and VLAN ID are inserted after the MAC source address, but before the original Ethertype/Length or Logical Link Control (LLC). The 1-bit CFI included a T-R Encapsulation bit so that Token Ring frames can be carried across Ethernet backbones without using 802.1H translation.

Frame Control Sequence Recomputation

The figure below shows how adding a tag in a frame recomputes the Frame Control Sequence. 802.1p and 802.1Q share the same tag.

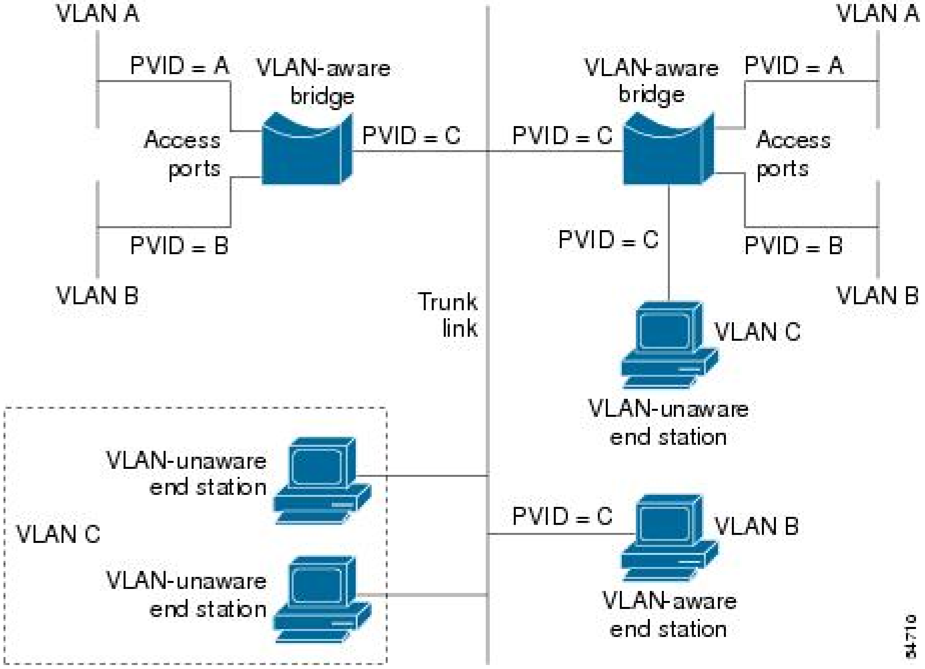
Figure 4: Adding a Tag Recomputes the Frame Control Sequence



Native VLAN

Each physical port has a parameter called PVID. Every 802.1Q port is assigned a PVID value that is of its native VLAN ID (default is VLAN 1). All untagged frames are assigned to the LAN specified in the PVID parameter. When a tagged frame is received by a port, the tag is respected. If the frame is untagged, the value contained in the PVID is considered as a tag. Because the frame is untagged and the PVID is tagged to allow the coexistence, as shown in the figure below, on the same pieces of cable of VLAN-aware bridge/stations and of VLAN-unaware bridges/stations. Consider, for example, the two stations connected to the central trunk link in the lower part of the figure below. They are VLAN-unaware and they will be associated to the VLAN C, because the PVIDs of the VLAN-aware bridges are equal to VLAN C. Because the VLAN-unaware stations will send only untagged frames, when the VLAN-aware bridge devices receive these untagged frames they will assign them to VLAN C.

Figure 5: Native VLAN



PVST+

PVST+ provides support for 802.1Q trunks and the mapping of multiple spanning trees to the single spanning tree of 802.1Q switches.

The PVST+ architecture distinguishes three types of regions:

- A PVST region
- A PVST+ region
- A MST region

Each region consists of a homogenous type of switch. A PVST region can be connected to a PVST+ region by connecting two ISL ports. Similarly, a PVST+ region can be connected to an MST region by connecting two 802.1Q ports.

At the boundary between a PVST region and a PVST+ region the mapping of spanning trees is one-to-one. At the boundary between a MST region and a PVST+ region, the ST in the MST region maps to one PVST in the PVST+ region. The one it maps to is called the common spanning tree (CST). The default CST is the PVST of VLAN 1 (Native VLAN).

All PVSTs, except for the CST, are tunneled through the MST region. Tunneling means that bridge protocol data units (BPDUs) are flooded through the MST region along the single spanning tree present in the MST region.

Ingress and Egress Rules

The BPDU transmission on the 802.1Q port of a PVST+ router will be implemented in compliance with the following rules:

- The CST BPDU (of VLAN 1, by default) is sent to the IEEE address.
- All the other BPDUs are sent to Shared Spanning Tree Protocol (SSTP)-Address and encapsulated with Logical Link Control-Subnetwork Access Protocol (LLC-SNAP) header.
- The BPDU of the CST and BPDU of the VLAN equal to the PVID of the 802.1Q trunk are sent untagged.
- All other BPDUs are sent tagged with the VLAN ID.
- The CST BPDU is also sent to the SSTP address.
- Each SSTP-addressed BPDU is also tailed by a Tag-Length-Value for the PVID checking.

The BPDU reception on the 802.1Q port of a PVST+ router will follow these rules:

- All untagged IEEE addressed BPDUs must be received on the PVID of the 802.1Q port.
- The IEEE addressed BPDUs whose VLAN ID matches the Native VLAN are processed by CST.
- All the other IEEE addressed BPDUs whose VLAN ID does not match the Native VLAN and whose port type is not of 802.1Q are processed by the spanning tree of that particular VLAN ID.
- The SSTP addressed BPDU whose VLAN ID is not equal to the TLV are dropped and the ports are blocked for inconsistency.
- All the other SSTP addressed BPDUs whose VLAN ID is not equal to the Native VLAN are processed by the spanning tree of that particular VLAN ID.
- The SSTP addressed BPDUs whose VLAN ID is equal to the Native VLAN are dropped. It is used for consistency checking.

Integrated Routing and Bridging

IRB enables a user to route a given protocol between routed interfaces and bridge groups or route a given protocol between the bridge groups. Integrated routing and bridging is supported on the following protocols:

- IP
- IPX
- AppleTalk

VLAN Colors

VLAN switching is accomplished through *frame tagging* where traffic originating and contained within a particular virtual topology carries a unique VLAN ID as it traverses a common backbone or trunk link. The VLAN ID enables VLAN switching devices to make intelligent forwarding decisions based on the embedded VLAN ID. Each VLAN is differentiated by a *color* , or VLAN identifier. The unique VLAN ID determines the *frame coloring* for the VLAN. Packets originating and contained within a particular VLAN carry the identifier that uniquely defines that VLAN (by the VLAN ID).

The VLAN ID allows VLAN switches and routers to selectively forward packets to ports with the same VLAN ID. The switch that receives the frame from the source station inserts the VLAN ID and the packet is switched onto the shared backbone network. When the frame exits the switched LAN, a switch strips the header and forwards the frame to interfaces that match the VLAN color. If you are using a Cisco network management product such as VlanDirector, you can actually color code the VLANs and monitor VLAN graphically.

Implementing VLANS

Network managers can logically group networks that span all major topologies, including high-speed technologies such as, ATM, FDDI, and Fast Ethernet. By creating virtual LANs, system and network administrators can control traffic patterns and react quickly to relocations and keep up with constant changes in the network due to moving requirements and node relocation just by changing the VLAN member list in the router configuration. They can add, remove, or move devices or make other changes to network configuration using software to make the changes.

Issues regarding creating VLANs should have been addressed when you developed your network design. Issues to consider include the following:

- Scalability
- Performance improvements
- Security
- Network additions, moves, and changes

Communication Between VLANs

Cisco IOS software provides full-feature routing at Layer 3 and translation at Layer 2 between VLANs. Five different protocols are available for routing between VLANs:

All five of these technologies are based on OSI Layer 2 bridge multiplexing mechanisms.

Inter-Switch Link Protocol

The Inter-Switch Link (ISL) protocol is used to interconnect two VLAN-capable Ethernet, Fast Ethernet, or Gigabit Ethernet devices, such as the Catalyst 3000 or 5000 switches and Cisco 7500 routers. The ISL protocol is a packet-tagging protocol that contains a standard Ethernet frame and the VLAN information associated with that frame. The packets on the ISL link contain a standard Ethernet, FDDI, or Token Ring frame and the VLAN information associated with that frame. ISL is currently supported only over Fast Ethernet links, but a single ISL link, or trunk, can carry different protocols from multiple VLANs.

Procedures for configuring ISL and Token Ring ISL (TRISL) features are provided in the Configuring Routing Between VLANs with Inter-Switch Link Encapsulation section.

IEEE 802.10 Protocol

The IEEE 802.10 protocol provides connectivity between VLANs. Originally developed to address the growing need for security within shared LAN/MAN environments, it incorporates authentication and encryption techniques to ensure data confidentiality and integrity throughout the network. Additionally, by functioning at Layer 2, it is well suited to high-throughput, low-latency switching environments. The IEEE 802.10 protocol can run over any LAN or HDLC serial interface.

Procedures for configuring routing between VLANs with IEEE 802.10 encapsulation are provided in the Configuring Routing Between VLANs with IEEE 802.10 section.

IEEE 802.1Q Protocol

The IEEE 802.1Q protocol is used to interconnect multiple switches and routers, and for defining VLAN topologies. Cisco currently supports IEEE 802.1Q for Fast Ethernet and Gigabit Ethernet interfaces.

Note Cisco does not support IEEE 802.1Q encapsulation for Ethernet interfaces.

Procedures for configuring routing between VLANs with IEEE 802.1Q encapsulation are provided in the Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation.

ATM LANE Protocol

The ATM LAN Emulation (LANE) protocol provides a way for legacy LAN users to take advantage of ATM benefits without requiring modifications to end-station hardware or software. LANE emulates a broadcast environment like IEEE 802.3 Ethernet on top of an ATM network that is a point-to-point environment.

LANE makes ATM function like a LAN. LANE allows standard LAN drivers like NDIS and ODI to be used. The virtual LAN is transparent to applications. Applications can use normal LAN functions without the underlying complexities of the ATM implementation. For example, a station can send broadcasts and multicasts, even though ATM is defined as a point-to-point technology and does not support any-to-any services.

To accomplish this, special low-level software is implemented on an ATM client workstation, called the LAN Emulation Client (LEC). The client software communicates with a central control point called a LAN Emulation Server (LES). A broadcast and unknown server (BUS) acts as a central point to distribute broadcasts and multicasts. The LAN Emulation Configuration Server (LECS) holds a database of LECs and the ELANs they belong to. The database is maintained by a network administrator.

These protocols are described in detail in the *Cisco Internetwork Design Guide* .

ATM LANE Fast Simple Server Replication Protocol

To improve the ATM LANE Simple Server Replication Protocol (SSRP), Cisco introduced the ATM LANE Fast Simple Server Replication Protocol (FSSRP). FSSRP differs from LANE SSRP in that all configured LANE servers of an ELAN are always active. FSSRP-enabled LANE clients have virtual circuits (VCs) established to a maximum of four LANE servers and BUSs at one time. If a single LANE server goes down, the LANE client quickly switches over to the next LANE server and BUS, resulting in no data or LE ARP table entry loss and no extraneous signalling.

The FSSRP feature improves upon SSRP such that LANE server and BUS switchover for LANE clients is immediate. With SSRP, a LANE server would go down, and depending on the network load, it may have taken considerable time for the LANE client to come back up joined to the correct LANE server and BUS. In addition to going down with SSRP, the LANE client would do the following:

- Clear out its data direct VCs
- Clear out its LE ARP entries
- Cause substantial signalling activity and data loss

FSSRP was designed to alleviate these problems with the LANE client. With FSSRP, each LANE client is simultaneously joined to up to four LANE servers and BUSs. The concept of the master LANE server and BUS is maintained; the LANE client uses the master LANE server when it needs LANE server BUS services. However, the difference between SSRP and FSSRP is that if and when the master LANE server goes down, the LANE client is already connected to multiple backup LANE servers and BUSs. The LANE client simply uses the next backup LANE server and BUS as the master LANE server and BUS.

VLAN Interoperability

Cisco IOS features bring added benefits to the VLAN technology. Enhancements to ISL, IEEE 802.10, and ATM LANE implementations enable routing of all major protocols between VLANs. These enhancements allow users to create more robust networks incorporating VLAN configurations by providing communications capabilities between VLANs.

Inter-VLAN Communications

The Cisco IOS supports full routing of several protocols over ISL and ATM LANE VLANs. IP, Novell IPX, and AppleTalk routing are supported over IEEE 802.10 VLANs. Standard routing attributes such as network advertisements, secondaries, and help addresses are applicable, and VLAN routing is fast switched. The table below shows protocols supported for each VLAN encapsulation format and corresponding Cisco IOS software releases in which support was introduced.

Table 5: Inter-VLAN Routing Protocol Support

Protocol	ISL	ATM LANE	IEEE 802.10
IP	Release 11.1	Release 10.3	Release 11.1
Novell IPX (default encapsulation)	Release 11.1	Release 10.3	Release 11.1
Novell IPX (configurable encapsulation)	Release 11.3	Release 10.3	Release 11.3
AppleTalk Phase II	Release 11.3	Release 10.3	--
DECnet	Release 11.3	Release 11.0	--

Protocol	ISL	ATM LANE	IEEE 802.10
Banyan VINES	Release 11.3	Release 11.2	--
XNS	Release 11.3	Release 11.2	--
CLNS	Release 12.1	--	--
IS-IS	Release 12.1	--	--

VLAN Translation

VLAN translation refers to the ability of the Cisco IOS software to translate between different VLANs or between VLAN and non-VLAN encapsulating interfaces at Layer 2. Translation is typically used for selective inter-VLAN switching of nonroutable protocols and to extend a single VLAN topology across hybrid switching environments. It is also possible to bridge VLANs on the main interface; the VLAN encapsulating header is preserved. Topology changes in one VLAN domain do not affect a different VLAN.

Designing Switched VLANs

By the time you are ready to configure routing between VLANs, you will have already defined them through the switches in your network. Issues related to network design and VLAN definition should be addressed during your network design. See the *Cisco Internetwork Design Guide* and the appropriate switch documentation for information on these topics:

- Sharing resources between VLANs
- Load balancing
- Redundant links
- Addressing
- Segmenting networks with VLANs--Segmenting the network into broadcast groups improves network security. Use router access lists based on station addresses, application types, and protocol types.
- Routers and their role in switched networks--In switched networks, routers perform broadcast management, route processing, and distribution, and provide communication between VLANs. Routers provide VLAN access to shared resources and connect to other parts of the network that are either logically segmented with the more traditional subnet approach or require access to remote sites across wide-area links.

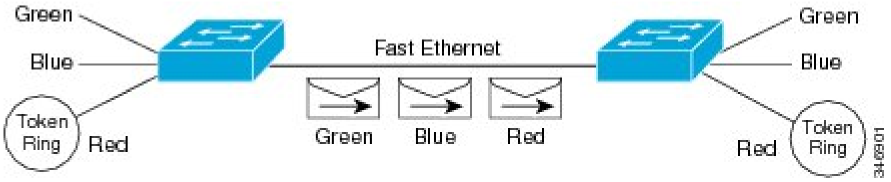
Frame Tagging in ISL

ISL is a Cisco protocol for interconnecting multiple switches and maintaining VLAN information as traffic goes between switches. ISL provides VLAN capabilities while maintaining full wire speed performance on Fast Ethernet links in full- or half-duplex mode. ISL operates in a point-to-point environment and will support up to 1000 VLANs. You can define virtually as many logical networks as are necessary for your environment.

With ISL, an Ethernet frame is encapsulated with a header that transports VLAN IDs between switches and routers. A 26-byte header that contains a 10-bit VLAN ID is propounded to the Ethernet frame.

A VLAN ID is added to the frame only when the frame is prepended for a nonlocal network. The figure below shows VLAN packets traversing the shared backbone. Each VLAN packet carries the VLAN ID within the packet header.

Figure 6: VLAN Packets Traversing the Shared Backbone



You can configure routing between any number of VLANs in your network. This section documents the configuration tasks for each protocol supported with ISL encapsulation. The basic process is the same, regardless of the protocol being routed. It involves the following tasks:

- Enabling the protocol on the router
- Enabling the protocol on the interface
- Defining the encapsulation format as ISL or TRISL
- Customizing the protocol according to the requirements for your environment

IEEE 802.1Q-in-Q VLAN Tag Termination on Subinterfaces

IEEE 802.1Q-in-Q VLAN Tag Termination simply adds another layer of IEEE 802.1Q tag (called “metro tag” or “PE-VLAN”) to the 802.1Q tagged packets that enter the network. The purpose is to expand the VLAN space by tagging the tagged packets, thus producing a “double-tagged” frame. The expanded VLAN space allows the service provider to provide certain services, such as Internet access on specific VLANs for specific customers, and yet still allows the service provider to provide other types of services for their other customers on other VLANs.

Generally the service provider’s customers require a range of VLANs to handle multiple applications. Service providers can allow their customers to use this feature to safely assign their own VLAN IDs on subinterfaces because these subinterface VLAN IDs are encapsulated within a service-provider designated VLAN ID for that customer. Therefore there is no overlap of VLAN IDs among customers, nor does traffic from different customers become mixed. The double-tagged frame is “terminated” or assigned on a subinterface with an expanded **encapsulation dot1q** command that specifies the two VLAN ID tags (outer VLAN ID and inner VLAN ID) terminated on the subinterface. See the figure below.

IEEE 802.1Q-in-Q VLAN Tag Termination is generally supported on whichever Cisco IOS features or protocols are supported on the subinterface; the exception is that Cisco 10000 series Internet router only supports PPPoE. For example if you can run PPPoE on the subinterface, you can configure a double-tagged frame for PPPoE. The only restriction is whether you assign ambiguous or unambiguous subinterfaces for the inner VLAN ID. See the figure below.

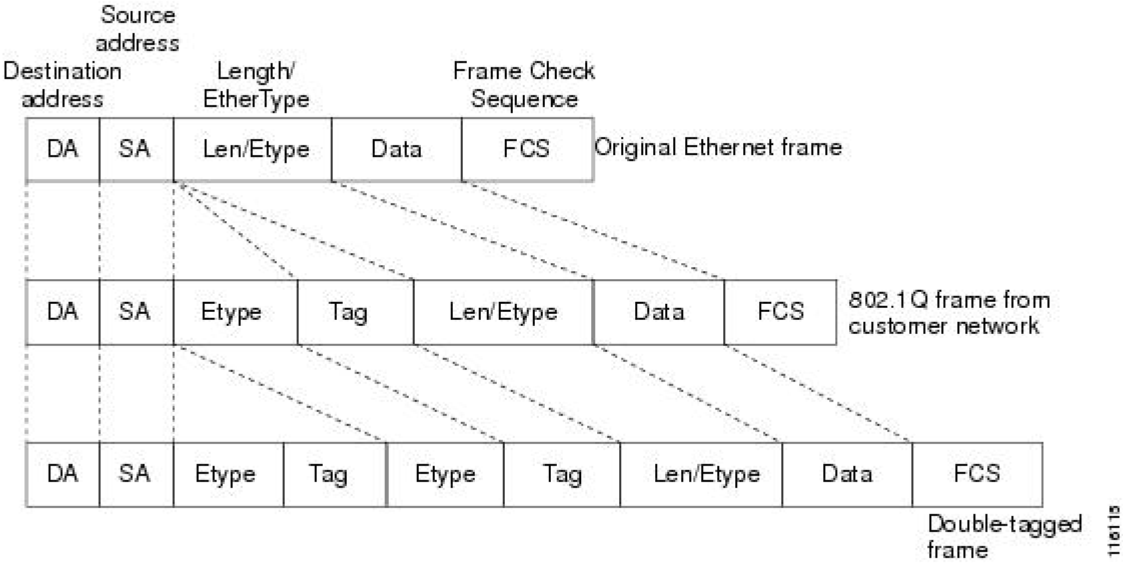
Note The Cisco 10000 series Internet router only supports Point-to-Point Protocol over Ethernet (PPPoE) and IP packets that are double-tagged for Q-in-Q VLAN tag termination. Specifically PPPoEoQ-in-Q and IPoQ-in-Q are supported.

The primary benefit for the service provider is reduced number of VLANs supported for the same number of customers. Other benefits of this feature include:

- PPPoE scalability. By expanding the available VLAN space from 4096 to approximately 16.8 million (4096 times 4096), the number of PPPoE sessions that can be terminated on a given interface is multiplied.
- When deploying Gigabyte Ethernet DSL Access Multiplexer (DSLAM) in wholesale model, you can assign the inner VLAN ID to represent the end-customer virtual circuit (VC) and assign the outer VLAN ID to represent the service provider ID.

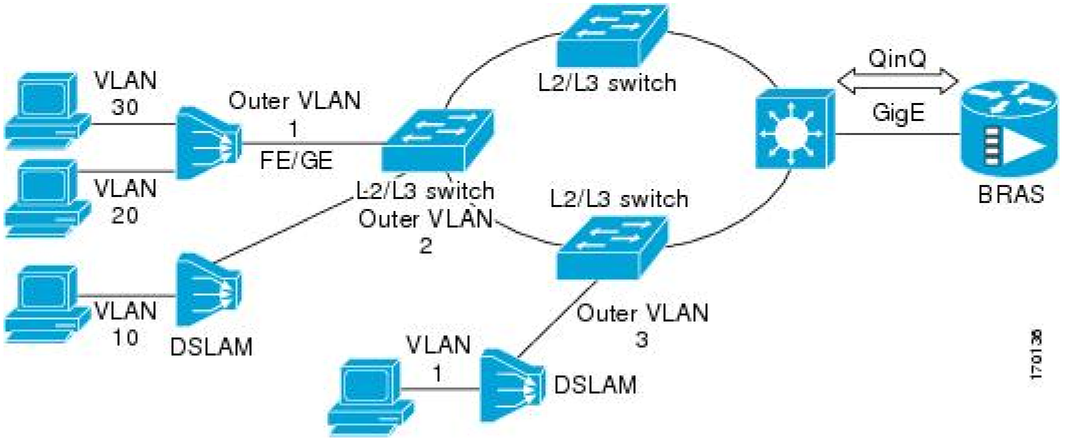
The Q-in-Q VLAN tag termination feature is simpler than the IEEE 802.1Q tunneling feature deployed for the Catalyst 6500 series switches or the Catalyst 3550 and Catalyst 3750 switches. Whereas switches require IEEE 802.1Q tunnels on interfaces to carry double-tagged traffic, routers need only encapsulate Q-in-Q VLAN tags within another level of 802.1Q tags in order for the packets to arrive at the correct destination as shown in figure below.

Figure 7: Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



Cisco 10000 Series Internet Router Application

For the emerging broadband Ethernet-based DSLAM market, the Cisco 10000 series Internet router supports Q-in-Q encapsulation. With the Ethernet-based DSLAM model shown in the figure below, customers typically get their own VLAN and all these VLANs are aggregated on a DSLAM.

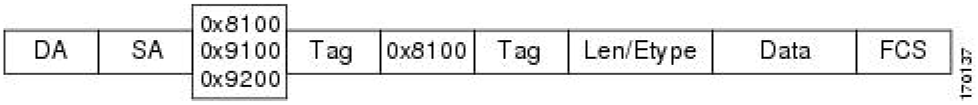


VLAN aggregation on a DSLAM will result in a lot of aggregate VLANs that at some point need to be terminated on the broadband remote access servers (BRAS). Although the model could connect the DSLAMs directly to the BRAS, a more common model uses the existing Ethernet-switched network where each DSLAM VLAN ID is tagged with a second tag (Q-in-Q) as it connects into the Ethernet-switched network.

The only model that is supported is PPPoE over Q-in-Q (PPPoEoQinQ). This can either be a PPP terminated session or as a L2TP LAC session.

The Cisco 10000 series Internet router already supports plain PPPoE and PPP over 802.1Q encapsulation. Supporting PPP over Q-in-Q encapsulation is new. PPP over Q-in-Q encapsulation processing is an extension to 802.1q encapsulation processing. A Q-in-Q frame looks like a VLAN 802.1Q frame, only it has two 802.1Q tags instead of one.

PPP over Q-in-Q encapsulation supports configurable outer tag Ethertype. The configurable Ethertype field values are 0x8100 (default), 0x9100, and 0x9200. See the figure below.



Security ACL Application on the Cisco 10000 Series Internet Router

The IEEE 802.1Q-in-Q VLAN Tag Termination feature provides limited security access control list (ACL) support for the Cisco 10000 series Internet router.

If you apply an ACL to PPPoE traffic on a Q-in-Q subinterface in a VLAN, apply the ACL directly on the PPPoE session, using virtual access interfaces (VAIs) or RADIUS attribute 11 or 242.

You can apply ACLs to virtual access interfaces by configuring them under virtual template interfaces. You can also configure ACLs by using RADIUS attribute 11 or 242. When you use attribute 242, a maximum of 30,000 sessions can have ACLs.

ACLs that are applied to the VLAN Q-in-Q subinterface have no effect and are silently ignored. In the following example, ACL 1 that is applied to the VLAN Q-in-Q subinterface level will be ignored:

```
Router(config)# interface FastEthernet3/0/0.100
Router(config-subif)# encapsulation dot1q 100 second-dot1q 200
Router(config-subif)# ip access-group 1
```

Unambiguous and Ambiguous Subinterfaces

The **encapsulation dot1q** command is used to configure Q-in-Q termination on a subinterface. The command accepts an Outer VLAN ID and one or more Inner VLAN IDs. The outer VLAN ID always has a specific value, while inner VLAN ID can either be a specific value or a range of values.

A subinterface that is configured with a single Inner VLAN ID is called an unambiguous Q-in-Q subinterface. In the following example, Q-in-Q traffic with an Outer VLAN ID of 101 and an Inner VLAN ID of 1001 is mapped to the Gigabit Ethernet 1/0.100 subinterface:

```
Router(config)# interface gigabitEehernet1/0.100
Router(config-subif)# encapsulation dot1q 101 second-dot1q 1001
```

A subinterface that is configured with multiple Inner VLAN IDs is called an ambiguous Q-in-Q subinterface. By allowing multiple Inner VLAN IDs to be grouped together, ambiguous Q-in-Q subinterfaces allow for a smaller configuration, improved memory usage and better scalability.

In the following example, Q-in-Q traffic with an Outer VLAN ID of 101 and Inner VLAN IDs anywhere in the 2001-2100 and 3001-3100 range is mapped to the Gigabit Ethernet 1/0.101 subinterface.:

```
Router(config)# interface gigabitethernet1/0.101
Router(config-subif)# encapsulation dot1q 101 second-dot1q 2001-2100,3001-3100
```

Ambiguous subinterfaces can also use the **any**keyword to specify the inner VLAN ID.

See the Monitoring and Maintaining VLAN Subinterfaces section for an example of how VLAN IDs are assigned to subinterfaces, and for a detailed example of how the **any** keyword is used on ambiguous subinterfaces.

Only PPPoE is supported on ambiguous subinterfaces. Standard IP routing is not supported on ambiguous subinterfaces.

Note On the Cisco 10000 series Internet router, Modular QoS services are only supported on unambiguous subinterfaces.

How to Configure Routing Between VLANS

Configuring a VLAN Range

Using the VLAN Range feature, you can group VLAN subinterfaces together so that any command entered in a group applies to every subinterface within the group. This capability simplifies configurations and reduces command parsing.

The VLAN Range feature provides the following benefits:

- Simultaneous Configurations: Identical commands can be entered once for a range of subinterfaces, rather than being entered separately for each subinterface.
- Overlapping Range Configurations: Overlapping ranges of subinterfaces can be configured.
- Customized Subinterfaces: Individual subinterfaces within a range can be customized or deleted.

Restrictions

- Each command you enter while you are in interface configuration mode with the **interface range** command is executed as it is entered. The commands are not batched together for execution after you exit interface configuration mode. If you exit interface configuration mode while the commands are being executed, some commands might not be executed on some interfaces in the range. Wait until the command prompt reappears before exiting interface configuration mode.
- The **no interface range** command is not supported. You must delete individual subinterfaces to delete a range.

Configuring a Range of VLAN Subinterfaces

Use the following commands to configure a range of VLAN subinterfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface range** {{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**} *slot* / *interface* **.** *subinterface* **-**{{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**}*slot* / *interface* **.** *subinterface*}
4. **encapsulation dot1Q** *vlan-id*
5. **no shutdown**
6. **exit**
7. **show running-config**
8. **show interfaces**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface range** {{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**} *slot* / *interface* **.** *subinterface*	Selects the range of subinterfaces to be configured. **Note** The spaces around the dash are required. For example, the command **interface range fastethernet 1 - 5**is valid; the command **interface range fastethernet 1-5** is not valid.

	Command or Action	Purpose
	-{{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**}*slot* / *interface* **.** *subinterface*} **Example:** `Router(config)# interface range fastethernet5/1.1 - fastethernet5/1.4`	
Step 4	**encapsulation dot1Q** *vlan-id* **Example:** `Router(config-if)# encapsulation dot1Q 301`	Applies a unique VLAN ID to each subinterface within the range. • *vlan-id* --Virtual LAN identifier. The allowed range is from 1 to 4095. • The VLAN ID specified by the *vlan-id* argument is applied to the first subinterface in the range. Each subsequent interface is assigned a VLAN ID, which is the specified *vlan-id* plus the subinterface number minus the first subinterface number (VLAN ID + subinterface number - first subinterface number).
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Activates the interface. • This command is required only if you shut down the interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies subinterface configuration.
Step 8	**show interfaces** **Example:** `Router# show interfaces`	Verifies that subinterfaces have been created.

Configuring Routing Between VLANs with Inter-Switch Link Encapsulation

This section describes the Inter-Switch Link (ISL) protocol and provides guidelines for configuring ISL and Token Ring ISL (TRISL) features. This section contains the following:

Configuring AppleTalk Routing over ISL

AppleTalk can be routed over VLAN subinterfaces using the ISL and IEEE 802.10 VLAN encapsulation protocols. The AppleTalk Routing over ISL and IEEE 802.10 Virtual LANs feature provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

To route AppleTalk over ISL or IEEE 802.10 between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform the steps in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appletalk routing** [**eigrp** *router-number*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **appletalk cable-range** *cable-range* [*network* **.** *node*]
7. **appletalk zone** *zone-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appletalk routing** [**eigrp** *router-number*] **Example:** `Router(config)# appletalk routing`	Enables AppleTalk routing globally on either ISL or 802.10 interfaces.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface Fddi 1/0.100`	Specifies the subinterface the VLAN will use.

	Command or Action	Purpose

Step 5	**encapsulation isl** *vlan-identifier*
	Example:
	Example:
	`or`
	Example:
	encapsulation sde *said*
	Example:
	`Router(config-if)# encapsulation sde 100`
		Defines the encapsulation format as either ISL (**isl**) or IEEE 802.10 (**sde**), and specifies the VLAN identifier or security association identifier, respectively.

Step 6	**appletalk cable-range** *cable-range* [*network* **.** *node*] **Example:** `Router(config-if)# appletalk cable-range 100-100 100.2`	Assigns the AppleTalk cable range and zone for the subinterface.
Step 7	**appletalk zone** *zone-name* **Example:** `Router(config-if)# appletalk zone 100`	Assigns the AppleTalk zone for the subinterface.

Configuring Banyan VINES Routing over ISL

Banyan VINES can be routed over VLAN subinterfaces using the ISL encapsulation protocol. The Banyan VINES Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software Banyan VINES support on a per-VLAN basis, allowing standard Banyan VINES capabilities to be configured on VLANs.

To route Banyan VINES over ISL between VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps in the following task in the order in which they appear:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vines routing** [*address*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **vines metric** [*whole* [*fraction*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vines routing** [*address*] **Example:** `Router(config)# vines routing`	Enables Banyan VINES routing globally.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used.
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 200`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 6	**vines metric** [*whole* [*fraction*]] **Example:** `Router(config-if)#vines metric 2`	Enables VINES routing metric on an interface.

Configuring DECnet Routing over ISL

DECnet can be routed over VLAN subinterfaces using the ISL VLAN encapsulation protocols. The DECnet Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software DECnet support on a per-VLAN basis, allowing standard DECnet capabilities to be configured on VLANs.

To route DECnet over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **decnet**[*network-number*] **routing**[*decnet-address*]
4. **interface** *type slot* / *port* . *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **decnet cost** [*cost-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **decnet**[*network-number*] **routing**[*decnet-address*] **Example:** `Router(config)# decnet routing 2.1`	Enables DECnet on the router.
Step 4	**interface** *type slot* / *port* . *subinterface-number* **Example:** `Router(config)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used.

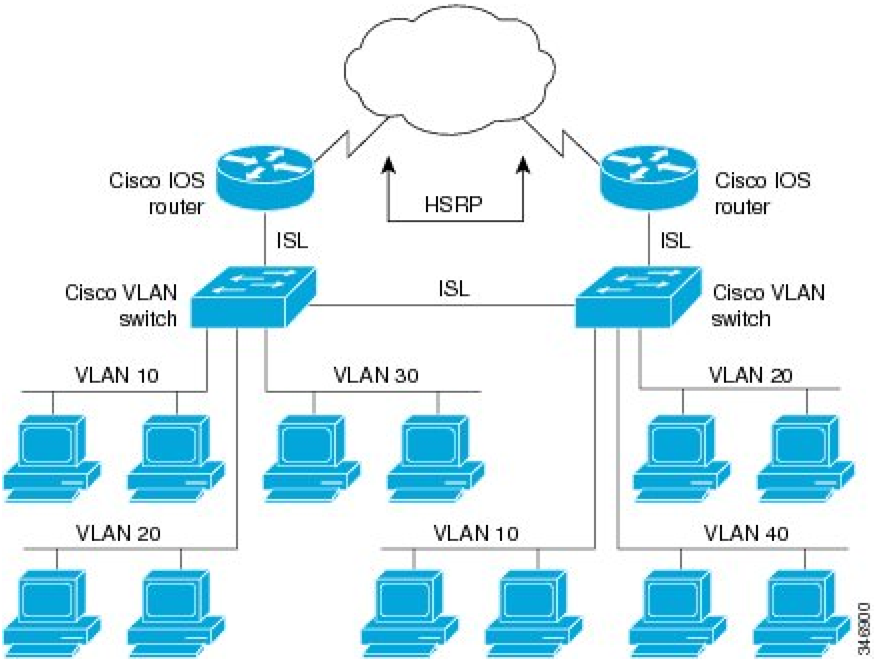
	Command or Action	Purpose
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 200`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 6	**decnet cost** [*cost-value*] **Example:** `Router(config-if)# decnet cost 4`	Enables DECnet cost metric on an interface.

Configuring the Hot Standby Router Protocol over ISL

The Hot Standby Router Protocol (HSRP) provides fault tolerance and enhanced routing performance for IP networks. HSRP allows Cisco IOS routers to monitor each other's operational status and very quickly assume packet forwarding responsibility in the event the current forwarding device in the HSRP group fails or is taken down for maintenance. The standby mechanism remains transparent to the attached hosts and can be deployed on any LAN type. With multiple Hot Standby groups, routers can simultaneously provide redundant backup and perform loadsharing across different IP subnets.

The figure below illustrates HSRP in use with ISL providing routing between several VLANs.

Figure 8: Hot Standby Router Protocol in VLAN Configurations



A separate HSRP group is configured for each VLAN subnet so that Cisco IOS router A can be the primary and forwarding router for VLANs 10 and 20. At the same time, it acts as backup for VLANs 30 and 40. Conversely, Router B acts as the primary and forwarding router for ISL VLANs 30 and 40, as well as the secondary and backup router for distributed VLAN subnets 10 and 20.

Running HSRP over ISL allows users to configure redundancy between multiple routers that are configured as front ends for VLAN IP subnets. By configuring HSRP over ISLs, users can eliminate situations in which a single point of failure causes traffic interruptions. This feature inherently provides some improvement in overall networking resilience by providing load balancing and redundancy capabilities between subnets and VLANs.

To configure HSRP over ISLs between VLANs, you need to create the environment in which it will be used. Perform the tasks described in the following sections in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port* **.** *subinterface-number*
4. **encapsulation isl** *vlan-identifier*
5. **ip address** *ip-address mask* [**secondary**]
6. Router(config-if)# **standby** [*group-number*] **ip**[*ip-address*[**secondary**]]
7. **standby** [*group-number*] **timers** *hellotime holdtime*
8. **standby** [*group-number*] **priority** *priority*
9. **standby** [*group-number*] **preempt**
10. **standby** [*group-number*] **track** *type-number*[*interface-priority*]
11. **standby** [*group-number*] **authentication** *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface FastEthernet 1/1.110`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)#  encapsulation isl 110`	Defines the encapsulation format, and specifies the VLAN identifier.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.1.1.2 255.255.255.0`	Specifies the IP address for the subnet on which ISL will be used.
Step 6	Router(config-if)# **standby** [*group-number*] **ip**[*ip-address*[**secondary**]] **Example:** `Router(config-if)# standby 1 ip 10.1.1.101`	Enables HSRP.
Step 7	**standby** [*group-number*] **timers** *hellotime holdtime* **Example:** `Router(config-if)# standby 1 timers 10 10`	Configures the time between hello packets and the hold time before other routers declare the active router to be down.
Step 8	**standby** [*group-number*] **priority** *priority* **Example:** `Router(config-if)# standby 1 priority 105`	Sets the Hot Standby priority used to choose the active router.
Step 9	**standby** [*group-number*] **preempt** **Example:** `Router(config-if)# standby 1 priority 105`	Specifies that if the local router has priority over the current active router, the local router should attempt to take its place as the active router.
Step 10	**standby** [*group-number*] **track** *type-number*[*interface-priority*] **Example:** `Router(config-if)# standby 1 track 4 5`	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the Hot Standby priority for the device is lowered.
Step 11	**standby** [*group-number*] **authentication** *string* **Example:** `Router(config-if)# standby 1 authentication hsrpword7`	Selects an authentication string to be carried in all HSRP messages.

What to Do Next

Note For more information on HSRP, see the “Configuring HSRP” module in the *Cisco IOS IP Application Services Configuration Guide* .

Configuring IP Routing over TRISL

The IP routing over TRISL VLANs feature extends IP routing capabilities to include support for routing IP frame types in VLAN configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **interface** *type slot* **/** *port* **.** *subinterface-number*
5. **encapsulation tr-isl trbrf-vlan** *vlanid* **bridge-num** *bridge-number*
6. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Router(config)# ip routing`	Enables IP routing on the router.

	Command or Action	Purpose
Step 4	**interface** *type slot* **/** *port* **.** *subinterface-number* **Example:** `Router(config)# interface FastEthernet4/0.1`	Specifies the subinterface on which TRISL will be used and enters interface configuration mode.
Step 5	**encapsulation tr-isl trbrf-vlan** *vlanid* **bridge-num** *bridge-number* **Example:** `Router(config-if# encapsulation tr-isl trbrf-vlan 999 bridge-num 14`	Defines the encapsulation for TRISL. • The DRiP database is automatically enabled when TRISL encapsulation is configured, and at least one TrBRF is defined, and the interface is configured for SRB or for routing with RIF.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if# ip address 10.5.5.1 255.255.255.0`	Sets a primary IP address for an interface. • A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a *subnet mask*. **Note** TRISL encapsulation must be specified for a subinterface before an IP address can be assigned to that subinterface.

Configuring IPX Routing on 802.10 VLANs over ISL

The IPX Encapsulation for 802.10 VLAN feature provides configurable IPX (Novell-FDDI, SAP, SNAP) encapsulation over 802.10 VLAN on router FDDI interfaces to connect the Catalyst 5000 VLAN switch. This feature extends Novell NetWare routing capabilities to include support for routing all standard IPX encapsulations for Ethernet frame types in VLAN configurations. Users with Novell NetWare environments can now configure any one of the three IPX Ethernet encapsulations to be routed using Secure Data Exchange (SDE) encapsulation across VLAN boundaries. IPX encapsulation options now supported for VLAN traffic include the following:

- Novell-FDDI (IPX FDDI RAW to 802.10 on FDDI)
- SAP (IEEE 802.2 SAP to 802.10 on FDDI)
- SNAP (IEEE 802.2 SNAP to 802.10 on FDDI)

NetWare users can now configure consolidated VLAN routing over a single VLAN trunking FDDI interface. Not all IPX encapsulations are currently supported for SDE VLAN. The IPX interior encapsulation support can be achieved by messaging the IPX header before encapsulating in the SDE format. Fast switching will also support all IPX interior encapsulations on non-MCI platforms (for example non-AGS+ and non-7000). With configurable Ethernet encapsulation protocols, users have the flexibility of using VLANs regardless of their NetWare Ethernet encapsulation. Configuring Novell IPX encapsulations on a per-VLAN basis facilitates migration between versions of Netware. NetWare traffic can now be routed across VLAN boundaries with standard encapsulation options (*arpa* , *sap* , and *snap*) previously unavailable. Encapsulation types and corresponding framing types are described in the "Configuring Novell IPX " module of the *Cisco IOS Novell IPX Configuration Guide* .

Note Only one type of IPX encapsulation can be configured per VLAN (subinterface). The IPX encapsulation used must be the same within any particular subnet; a single encapsulation must be used by all NetWare systems that belong to the same VLAN.

To configure Cisco IOS software on a router with connected VLANs to exchange different IPX framing protocols, perform the steps described in the following task in the order in which they are appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipx routing** [*node*]
4. **interface** *fddi slot* / *port* **.** *subinterface-number*
5. **encapsulation sde** *vlan-identifier*
6. **ipx network** *network* **encapsulation** *encapsulation-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipx routing** [*node*] **Example:** `Router(config)# ipx routing`	Enables IPX routing globally.
Step 4	**interface** *fddi slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface 2/0.1`	Specifies the subinterface on which SDE will be used and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation sde** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 20`	Defines the encapsulation format and specifies the VLAN identifier.
Step 6	**ipx network** *network* **encapsulation** *encapsulation-type* **Example:** `Router(config-if)# ipx network 20 encapsulation sap`	Specifies the IPX encapsulation among Novell-FDDI, SAP, or SNAP.

Configuring IPX Routing over TRISL

The IPX Routing over ISL VLANs feature extends Novell NetWare routing capabilities to include support for routing all standard IPX encapsulations for Ethernet frame types in VLAN configurations. Users with Novell NetWare environments can configure either SAP or SNAP encapsulations to be routed using the TRISL encapsulation across VLAN boundaries. The SAP (Novell Ethernet_802.2) IPX encapsulation is supported for VLAN traffic.

NetWare users can now configure consolidated VLAN routing over a single VLAN trunking interface. With configurable Ethernet encapsulation protocols, users have the flexibility of using VLANs regardless of their NetWare Ethernet encapsulation. Configuring Novell IPX encapsulations on a per-VLAN basis facilitates migration between versions of Netware. NetWare traffic can now be routed across VLAN boundaries with standard encapsulation options (*sap* and *snap*) previously unavailable. Encapsulation types and corresponding framing types are described in the "Configuring Novell IPX " module of the *Cisco IOS Novell IPX Configuration Guide* .

Note Only one type of IPX encapsulation can be configured per VLAN (subinterface). The IPX encapsulation used must be the same within any particular subnet: A single encapsulation must be used by all NetWare systems that belong to the same LANs.

To configure Cisco IOS software to exchange different IPX framing protocols on a router with connected VLANs, perform the steps in the following task in the order in which they are appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipx routing** [*node*]
4. **interface** *type slot* **/** *port* **.** *subinterface-number*
5. **encapsulation tr-isl trbrf-vlan** *trbrf-vlan* **bridge-num** *bridge-num*
6. **ipx network** *network* **encapsulation** *encapsulation-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipx routing** [*node*] **Example:** `Router(config)# source-bridge ring-group 100`	Enables IPX routing globally.
Step 4	**interface** *type slot* **/** *port* **.** *subinterface-number* **Example:** `Router(config)# interface TokenRing 3/1`	Specifies the subinterface on which TRISL will be used and enters interface configuration mode.
Step 5	**encapsulation tr-isl trbrf-vlan** *trbrf-vlan* **bridge-num** *bridge-num* **Example:** `Router(config-if)#encapsulation tr-isl trbrf-vlan 999 bridge-num 14`	Defines the encapsulation for TRISL.
Step 6	**ipx network** *network* **encapsulation** *encapsulation-type* **Example:** `Router(config-if)# ipx network 100 encapsulation sap`	Specifies the IPX encapsulation on the subinterface by specifying the NetWare network number (if necessary) and the encapsulation type.

What to Do Next

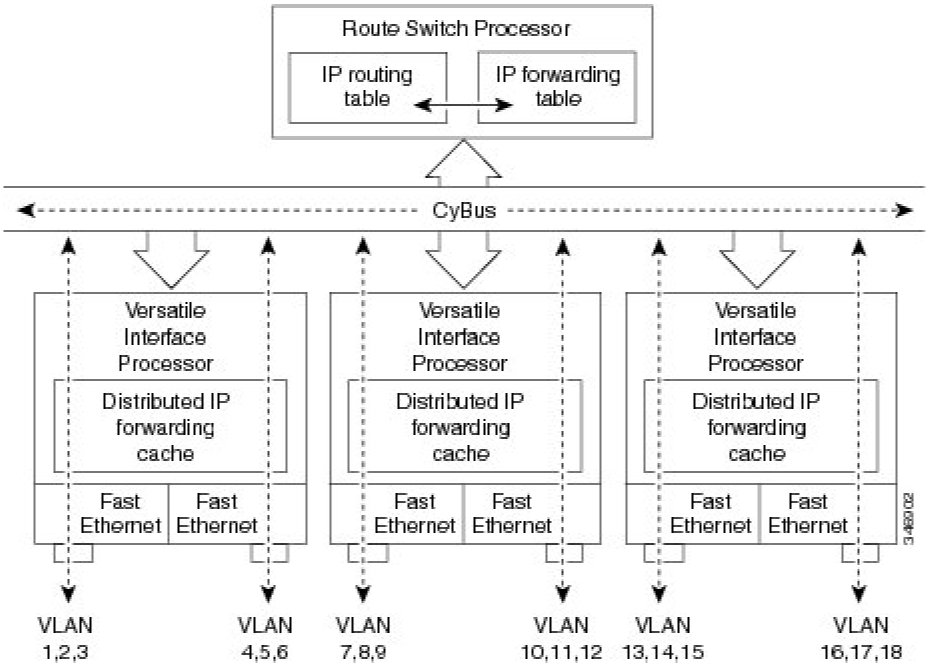
Note The default IPX encapsulation format for Cisco IOS routers is “novell-ether” (Novell Ethernet_802.3). If you are running Novell Netware 3.12 or 4.0, the new Novell default encapsulation format is Novell Ethernet_802.2 and you should configure the Cisco router with the IPX encapsulation format “sap.”

Configuring VIP Distributed Switching over ISL

With the introduction of the VIP distributed ISL feature, ISL encapsulated IP packets can be switched on Versatile Interface Processor (VIP) controllers installed on Cisco 7500 series routers.

The second generation VIP2 provides distributed switching of IP encapsulated in ISL in VLAN configurations. Where an aggregation route performs inter-VLAN routing for multiple VLANs, traffic can be switched autonomously on-card or between cards rather than through the central Route Switch Processor (RSP). The figure below shows the VIP distributed architecture of the Cisco 7500 series router.

Figure 9: Cisco 7500 Distributed Architecture



This distributed architecture allows incremental capacity increases by installation of additional VIP cards. Using VIP cards for switching the majority of IP VLAN traffic in multiprotocol environments substantially increases routing performance for the other protocols because the RSP offloads IP and can then be dedicated to switching the non-IP protocols.

VIP distributed switching offloads switching of ISL VLAN IP traffic to the VIP card, removing involvement from the main CPU. Offloading ISL traffic to the VIP card substantially improves networking performance. Because you can install multiple VIP cards in a router, VLAN routing capacity is increased linearly according to the number of VIP cards installed in the router.

To configure distributed switching on the VIP, you must first configure the router for IP routing. Perform the tasks described below in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **interface** *type slot* / *port-adapter* / *port*
5. **ip route-cache distributed**
6. **encapsulation isl** *vlan-identifier*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Router(config)# ip routing`	Enables IP routing on the router. • For more information about configuring IP routing, see the appropriate Cisco IOS *IP Routing Configuration Guide* for the version of Cisco IOS you are using.
Step 4	**interface** *type slot* / *port-adapter* / *port* **Example:** `Router(config)# interface FastEthernet1/0/0`	Specifies the interface and enters interface configuration mode.
Step 5	**ip route-cache distributed** **Example:** `Router(config-if)# ip route-cache distributed`	Enables VIP distributed switching of IP packets on the interface.
Step 6	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 1`	Defines the encapsulation format as ISL, and specifies the VLAN identifier.

Configuring XNS Routing over ISL

XNS can be routed over VLAN subinterfaces using the ISL VLAN encapsulation protocol. The XNS Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software XNS support on a per-VLAN basis, allowing standard XNS capabilities to be configured on VLANs.

To route XNS over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **xns routing** [*address*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **xns network** [*number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**xns routing** [*address*] **Example:** `Router(config)# xns routing 0123.4567.adcb`	Enables XNS routing globally.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 100`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.

	Command or Action	Purpose
Step 6	**xns network** [*number*] **Example:** `Router(config-if)# xns network 20`	Enables XNS routing on the subinterface.

Configuring CLNS Routing over ISL

CLNS can be routed over VLAN subinterfaces using the ISL VLAN encapsulation protocol. The CLNS Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software CLNS support on a per-VLAN basis, allowing standard CLNS capabilities to be configured on VLANs.

To route CLNS over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clns routing**
4. **interface** *type slot* **/** *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **clns enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**clns routing** **Example:** `Router(config)# clns routing`	Enables CLNS routing globally.

	Command or Action	Purpose
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config-if)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 100`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 6	**clns enable** **Example:** `Router(config-if)# clns enable`	Enables CLNS routing on the subinterface.

Configuring IS-IS Routing over ISL

IS-IS routing can be enabled over VLAN subinterfaces using the ISL VLAN encapsulation protocol. The IS-IS Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software IS-IS support on a per-VLAN basis, allowing standard IS-IS capabilities to be configured on VLANs.

To enable IS-IS over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*tag*]
4. **net** *network-entity-title*
5. **interface** *type slot* / *port* **.** *subinterface-number*
6. **encapsulation isl** *vlan-identifier*
7. **clns router isis network** [*tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*tag*] **Example:** `Router(config)# isis routing test-proc2`	Enables IS-IS routing, and enters router configuration mode.
Step 4	**net** *network-entity-title* **Example:** `Router(config)# net 49.0001.0002.aaaa.aaaa.aaaa.00`	Configures the NET for the routing process.
Step 5	**interface** *type slot* **/** *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 2.`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.
Step 6	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 101`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 7	**clns router isis network** [*tag*] **Example:** `Router(config-if)# clns router is-is network test-proc2`	Specifies the interfaces that should be actively routing IS-IS.

Configuring Routing Between VLANs with IEEE 802.10 Encapsulation

This section describes the required and optional tasks for configuring routing between VLANs with IEEE 802.10 encapsulation.

HDLC serial links can be used as VLAN trunks in IEEE 802.10 VLANs to extend a virtual topology beyond a LAN backbone.

AppleTalk can be routed over VLAN subinterfaces using the ISL or IEEE 802.10 VLANs feature that provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

AppleTalk users can now configure consolidated VLAN routing over a single VLAN trunking interface. Prior to introduction of this feature, AppleTalk could be routed only on the main interface on a LAN port. If AppleTalk routing was disabled on the main interface or if the main interface was shut down, the entire physical interface would stop routing any AppleTalk packets. With this feature enabled, AppleTalk routing on subinterfaces will be unaffected by changes in the main interface with the main interface in the “no-shut” state.

To route AppleTalk over IEEE 802.10 between VLANs, create the environment in which it will be used by customizing the subinterface and perform the tasks described in the following steps in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appletalk routing** [**eigrp** *router-number*]
4. **interface fastethernet** *slot* **/** *port* **.** subinterface-number
5. **appletalk cable-range** *cable-range* [*network* **.** *node*]
6. **appletalk zone** *>zone-name*
7. **encapsulation sde** *said*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appletalk routing** [**eigrp** *router-number*] **Example:** `Router(config)# appletalk routing`	Enables AppleTalk routing globally.

	Command or Action	Purpose
Step 4	**interface fastethernet** *slot* **/** *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.00`	Specifies the subinterface the VLAN will use and enters inerface configuration mode.
Step 5	**appletalk cable-range** *cable-range* [*network* **.** *node*] **Example:** `Router(config-if)# appletalk 100-100 100.1`	Assigns the AppleTalk cable range and zone for the subinterface.
Step 6	**appletalk zone** >*zone-name* **Example:** `Router(config-if)# appletalk zone eng`	Assigns the AppleTalk zone for the subinterface.
Step 7	**encapsulation sde** *said* **Example:** `Router(config-if)# encapsulation sde 100`	Defines the encapsulation format as IEEE 802.10 (sde) and specifies the VLAN identifier or security association identifier, respectively.

What to Do Next

Note For more information on configuring AppleTalk, see the “Configuring AppleTalk” module in the *Cisco IOS AppleTalk Configuration Guide* .

Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation

This section describes the required and optional tasks for configuring routing between VLANs with IEEE 802.1Q encapsulation. The IEEE 802.1Q protocol is used to interconnect multiple switches and routers, and for defining VLAN topologies.

Prerequisites

Configuring routing between VLANs with IEEE 802.1Q encapsulation assumes the presence of a single spanning tree and of an explicit tagging scheme with one-level tagging.

You can configure routing between any number of VLANs in your network.

Restrictions

The IEEE 802.1Q standard is extremely restrictive to untagged frames. The standard provides only a per-port VLANs solution for untagged frames. For example, assigning untagged frames to VLANs takes into consideration only the port from which they have been received. Each port has a parameter called a *permanent virtual identification* (Native VLAN) that specifies the VLAN assigned to receive untagged frames.

The main characteristics of the IEEE 802.1Q are that it assigns frames to VLANs by filtering and that the standard assumes the presence of a single spanning tree and of an explicit tagging scheme with one-level tagging.

This section contains the configuration tasks for each protocol supported with IEEE 802.1Q encapsulation. The basic process is the same, regardless of the protocol being routed. It involves the following tasks:

- Enabling the protocol on the router
- Enabling the protocol on the interface
- Defining the encapsulation format as IEEE 802.1Q
- Customizing the protocol according to the requirements for your environment

To configure IEEE 802.1Q on your network, perform the following tasks. One of the following tasks is required depending on the protocol being used.

- Configuring AppleTalk Routing over IEEE 802.1Q, on page 157 (required)
- Configuring IP Routing over IEEE 802.1Q, on page 159 (required)
- Configuring IPX Routing over IEEE 802.1Q, on page 160 (required)

The following tasks are optional. Perform the following tasks to connect a network of hosts over a simple bridging-access device to a remote access concentrator bridge between IEEE 802.1Q VLANs. The following sections contain configuration tasks for the Integrated Routing and Bridging, Transparent Bridging, and PVST+ Between VLANs with IEEE 802.1Q Encapsulation:

- Configuring a VLAN for a Bridge Group with Default VLAN1, on page 162 (optional)
- Configuring a VLAN for a Bridge Group as a Native VLAN, on page 163 (optional)

Configuring AppleTalk Routing over IEEE 802.1Q

AppleTalk can be routed over virtual LAN (VLAN) subinterfaces using the IEEE 802.1Q VLAN encapsulation protocol. AppleTalk Routing provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

To route AppleTalk over IEEE 802.1Q between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform the steps in the order in which they appear.

Use the following task to enable AppleTalk routing on IEEE 802.1Q interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appletalk routing** [**eigrp** *router-number*]
4. **interface fastethernet** *slot* / *port* **.** subinterface-number
5. **encapsulation dot1q** *vlan-identifier*
6. **appletalk cable-range** *cable-range* [*network* **.** *node*]
7. **appletalk zone** *zone-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appletalk routing** [**eigrp** *router-number*] **Example:** `Router(config)# appletalk routing`	Enables AppleTalk routing globally.
Step 4	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.00`	Specifies the subinterface the VLAN will use and enters interface configuration mode.
Step 5	**encapsulation dot1q** *vlan-identifier* **Example:** `Router(config-if)# encapsulation dot1q 100`	Defines the encapsulation format as IEEE 802.1Q (**dot1q**), and specifies the VLAN identifier.
Step 6	**appletalk cable-range** *cable-range* [*network* **.** *node*] **Example:** `Router(config-if)# appletalk cable-range 100-100 100.1`	Assigns the AppleTalk cable range and zone for the subinterface.

	Command or Action	Purpose
Step 7	**appletalk zone** *zone-name* **Example:** `Router(config-if)# appletalk zone eng`	Assigns the AppleTalk zone for the subinterface.

What to Do Next

Note For more information on configuring AppleTalk, see the "Configuring AppleTalk" module in the *Cisco IOS AppleTalk Configuration Guide* .

Configuring IP Routing over IEEE 802.1Q

IP routing over IEEE 802.1Q extends IP routing capabilities to include support for routing IP frame types in VLAN configurations using the IEEE 802.1Q encapsulation.

To route IP over IEEE 802.1Q between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform the tasks described in the following sections in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. ip routing
4. **interface fastethernet** *slot* **/** *port* **.** subinterface-number
5. **encapsulation dot1q** vlanid
6. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	ip routing **Example:** `Router(config)# ip routing`	Enables IP routing on the router.
Step 4	**interface fastethernet** *slot* **/** *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.101`	Specifies the subinterface on which IEEE 802.1Q will be used and enters interface configuration mode.
Step 5	**encapsulation dot1q** vlanid **Example:** `Router(config-if)# encapsulation dot1q 101`	Defines the encapsulation format at IEEE.802.1Q (dot1q) and specifies the VLAN identifier.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip addr 10.0.0.11 255.0.0.0`	Sets a primary IP address and mask for the interface.

What to Do Next

Once you have IP routing enabled on the router, you can customize the characteristics to suit your environment. See the appropriate *Cisco IOS IP Routing Configuration Guide* for the version of Cisco IOS you are using.

Configuring IPX Routing over IEEE 802.1Q

IPX routing over IEEE 802.1Q VLANs extends Novell NetWare routing capabilities to include support for routing Novell Ethernet_802.3 encapsulation frame types in VLAN configurations. Users with Novell NetWare environments can configure Novell Ethernet_802.3 encapsulation frames to be routed using IEEE 802.1Q encapsulation across VLAN boundaries.

To configure Cisco IOS software on a router with connected VLANs to exchange IPX Novell Ethernet_802.3 encapsulated frames, perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipx routing** [*node*]
4. **interface fastethernet** *slot* **/** *port* **.** subinterface-number
5. **encapsulation dot1q** vlanid
6. **ipx network** *network*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipx routing** [*node*] **Example:** `Router(config)# ipx routing`	Enables IPX routing globally.
Step 4	**interface fastethernet** *slot* **/** *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.102`	Specifies the subinterface on which IEEE 802.1Q will be used and enters interface configuration mode.
Step 5	**encapsulation dot1q** vlanid **Example:** `Router(config-if)# encapsulation dot1q 102`	Defines the encapsulation format at IEEE.802.1Q (**dot1q**) and specifies the VLAN identifier.
Step 6	**ipx network** *network* **Example:** `Router(config-if)# ipx network 100`	Specifies the IPX network number.

Configuring a VLAN for a Bridge Group with Default VLAN1

Use the following task to configure a VLAN associated with a bridge group with a default native VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *slot* **/** *port* **.** subinterface-number
4. **encapsulation dot1q** vlanid
5. **bridge-group** *bridge-group*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *slot* **/** *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.100`	Selects a particular interface to configure and enters interface configuration mode.
Step 4	**encapsulation dot1q** vlanid **Example:** `Router(config-subif)# encapsulation dot1q 1`	Defines the encapsulation format at IEEE.802.1Q (dot1q) and specifies the VLAN identifier. • The specified VLAN is by default the native VLAN. **Note** If there is no explicitly defined native VLAN, the default VLAN1 becomes the native VLAN.
Step 5	**bridge-group** *bridge-group* **Example:** `Router(config-subif)# bridge-group 1`	Assigns the bridge group to the interface.

Configuring a VLAN for a Bridge Group as a Native VLAN

Use the following task to configure a VLAN associated to a bridge group as a native VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *slot* **/** *port* **.** subinterface-number
4. **encapsulation dot1q** *vlanid* **native**
5. **bridge-group** *bridge-group*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *slot* **/** *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.100`	Selects a particular interface to configure and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlanid* **native** **Example:** `Router(config-subif)# encapsulation dot1q 20 native`	Defines the encapsulation format at IEEE.802.1Q (**dot1q**) and specifies the VLAN identifier. VLAN 20 is specified as the native VLAN. **Note** If there is no explicitly defined native VLAN, the default VLAN1 becomes the native VLAN.
Step 5	**bridge-group** *bridge-group* **Example:** `Router(config-subif)# bridge-group 1`	Assigns the bridge group to the interface.

What to Do Next

Note If there is an explicitly defined native VLAN, VLAN1 will only be used to process CST.

Configuring IEEE 802.1Q-in-Q VLAN Tag Termination

Encapsulating IEEE 802.1Q VLAN tags within 802.1Q enables service providers to use a single VLAN to support customers who have multiple VLANs. The IEEE 802.1Q-in-Q VLAN Tag Termination feature on the subinterface level preserves VLAN IDs and keeps traffic in different customer VLANs segregated.

You must have checked Feature Navigator to verify that your Cisco device and software image support this feature.

You must be connected to an Ethernet device that supports double VLAN tag imposition/disposition or switching.

The following restrictions apply to the Cisco 10000 series Internet router for configuring IEEE 802.1Q-in-Q VLAN tag termination:

- Supported on Ethernet, FastEthernet, or Gigabit Ethernet interfaces.
- Supports only Point-to-Point Protocol over Ethernet (PPPoE) packets that are double-tagged for Q-in-Q VLAN tag termination.
- IP and Multiprotocol Label Switching (MPLS) packets are not supported.
- Modular QoS can be applied to unambiguous subinterfaces only.
- Limited ACL support.

Perform these tasks to configure the main interface used for the Q-in-Q double tagging and to configure the subinterfaces.

Configuring EtherType Field for Outer VLAN Tag Termination

The following restrictions are applicable for the Cisco 10000 series Internet router:

- PPPoE is already configured.
- Virtual private dial-up network (VPDN) is enabled.

The first task is optional. A step in this task shows you how to configure the EtherType field to be 0x9100 for the outer VLAN tag, if that is required.

After the subinterface is defined, the 802.1Q encapsulation is configured to use the double tagging.

To configure the EtherType field for Outer VLAN Tag Termination, use the following steps. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dot1q tunneling ethertype** *ethertype*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**dot1q tunneling ethertype** *ethertype* **Example:** `Router(config-if)# dot1q tunneling ethertype 0x9100`	(Optional) Defines the Ethertype field type used by peer devices when implementing Q-in-Q VLAN tagging. • Use this command if the Ethertype of peer devices is 0x9100 or 0x9200 (0x9200 is only supported on the Cisco 10000 series Internet router). • Cisco 10000 series Internet router supports both the 0x9100 and 0x9200 Ethertype field types.

Configuring the Q-in-Q Subinterface

Use the following steps to configure Q-in-Q subinterfaces. This task is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* **.** *subinterface-number*
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*| *vlan-id* **-** *vlan-id* [**,** *vlan-id* **-** *vlan-id*]}
5. **pppoe enable** [**group** *group-name*]
6. **exit**
7. Repeat Step 3 to configure another subinterface.
8. Repeat Step 4 and Step 5 to specify the VLAN tags to be terminated on the subinterface.
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **.** *subinterface-number* **Example:** `Router(config)# interface gigabitethernet 1/0/0.1`	Configures a subinterface and enters subinterface configuration mode.

Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*| *vlan-id* **-** *vlan-id* [**,** *vlan-id* **-** *vlan-id*]}
	Example:
	`Router(config-subif)# encapsulation dot1q 100 second-dot1q 200`
		(Required) Enables the 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
		• Use the **second-dot1q** keyword and the *vlan-id*argument to specify the VLAN tags to be terminated on the subinterface.
		• In this example, an unambiguous Q-in-Q subinterface is configured because only one inner VLAN ID is specified.
		• Q-in-Q frames with an outer VLAN ID of 100 and an inner VLAN ID of 200 will be terminated.

Step 5	**pppoe enable** [**group** *group-name*]	Enables PPPoE sessions on a subinterface.

	Command or Action	Purpose
	Example: `Router(config-subif)# pppoe enable group vpn1`	- The example specifies that the PPPoE profile, vpn1, will be used by PPPoE sessions on the subinterface.
Step 6	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode and returns to interface configuration mode. - Repeat this step one more time to exit interface configuration mode.
Step 7	Repeat Step 3 to configure another subinterface. **Example:** `Router(config-if)# interface gigabitethernet 1/0/0.2`	(Optional) Configures a subinterface and enters subinterface configuration mode.

Step 8	Repeat Step 4 and Step 5 to specify the VLAN tags to be terminated on the subinterface.
	Example:
	`Router(config-subif)# encapsulation dot1q 100 second-dot1q 100-199,201-600`
	Example:
	Example:
	`Router(config-subif)# pppoe enable group vpn1`
	Example:
		Step 4 enables the 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
		- Use the **second-dot1q** keyword and the *vlan-id*argument to specify the VLAN tags to be terminated on the subinterface.
		- In the example, an ambiguous Q-in-Q subinterface is configured because a range of inner VLAN IDs is specified.
		- Q-in-Q frames with an outer VLAN ID of 100 and an inner VLAN ID in the range of 100 to 199 or 201 to 600 will be terminated.
		Step 5 enables PPPoE sessions on the subinterface. The example specifies that the PPPoE profile, vpn1, will be used by PPPoE sessions on the subinterface.
		Note Step 5 is required for the Cisco 10000 series Internet router because it only supports PPPoEoQinQ traffic.

Step 9	**end**
	Example:
	`Router(config-subif)# end`
		Exits subinterface configuration mode and returns to privileged EXEC mode.

Verifying the IEEE 802.1Q-in-Q VLAN Tag Termination

Perform this optional task to verify the configuration of the IEEE 802.1Q-in-Q VLAN Tag Termination feature.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **show vlans dot1q** [**internal** | *interface-type interface-number* **.***subinterface-number*[**detail**] | *outer-id*[*interface-type interface-number* | **second-dot1q** [*inner-id*| **any**]] [**detail**]]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config**

Use this command to show the currently running configuration on the device. You can use delimiting characters to display only the relevant parts of the configuration.

The following shows the currently running configuration on a Cisco 7300 series router:

Example:

```
Router# show running-config
.
.
.
interface FastEthernet0/0.201
 encapsulation dot1Q 201
 ip address 10.7.7.5 255.255.255.252
!
interface FastEthernet0/0.401
 encapsulation dot1Q 401
 ip address 10.7.7.13 255.255.255.252
!
interface FastEthernet0/0.201999
 encapsulation dot1Q 201 second-dot1q any
 pppoe enable
!
interface FastEthernet0/0.2012001
 encapsulation dot1Q 201 second-dot1q 2001
 ip address 10.8.8.9 255.255.255.252
!
interface FastEthernet0/0.2012002
 encapsulation dot1Q 201 second-dot1q 2002
 ip address 10.8.8.13 255.255.255.252
!
interface FastEthernet0/0.4019999
 encapsulation dot1Q 401 second-dot1q 100-900,1001-2000
 pppoe enable
!
interface GigabitEthernet5/0.101
 encapsulation dot1Q 101
 ip address 10.7.7.1 255.255.255.252
!
interface GigabitEthernet5/0.301
 encapsulation dot1Q 301
 ip address 10.7.7.9 255.255.255.252
!
```

```
interface GigabitEthernet5/0.301999
 encapsulation dot1Q 301 second-dot1q any
 pppoe enable
!
interface GigabitEthernet5/0.1011001
 encapsulation dot1Q 101 second-dot1q 1001
 ip address 10.8.8.1 255.255.255.252
!
interface GigabitEthernet5/0.1011002
 encapsulation dot1Q 101 second-dot1q 1002
 ip address 10.8.8.5 255.255.255.252
!
interface GigabitEthernet5/0.1019999
 encapsulation dot1Q 101 second-dot1q 1-1000,1003-2000
 pppoe enable
.
.
.
```

The following shows the currently running configuration on a Cisco 10000 series Internet router:

Example:

```
Router# show running-config
.
.
.
interface FastEthernet1/0/0.201
 encapsulation dot1Q 201
 ip address 10.7.7.5 255.255.255.252
!
interface FastEthernet1/0/0.401
 encapsulation dot1Q 401
 ip address 10.7.7.13 255.255.255.252
!
interface FastEthernet1/0/0.201999
 encapsulation dot1Q 201 second-dot1q any
 pppoe enable
!
interface FastEthernet1/0/0.4019999
 encapsulation dot1Q 401 second-dot1q 100-900,1001-2000
 pppoe enable
!
interface GigabitEthernet5/0/0.101
 encapsulation dot1Q 101
 ip address 10.7.7.1 255.255.255.252
!
interface GigabitEthernet5/0/0.301
 encapsulation dot1Q 301
 ip address 10.7.7.9 255.255.255.252
!
interface GigabitEthernet5/0/0.301999
 encapsulation dot1Q 301 second-dot1q any
 pppoe enable
!
interface GigabitEthernet5/0/0.1019999
 encapsulation dot1Q 101 second-dot1q 1-1000,1003-2000
 pppoe enable
.
.
.
```

Step 3 **show vlans dot1q** [**internal** | *interface-type interface-number* **.***subinterface-number*[**detail**] | *outer-id*[*interface-type interface-number* | **second-dot1q** [*inner-id*| **any**]] [**detail**]]

Use this command to show the statistics for all the 802.1Q VLAN IDs. In this example, only the outer VLAN ID is displayed.

Note The **show vlans dot1q**command is not supported on the Cisco 10000 series Internet router.

Example:

```
Router# show vlans dot1q
Total statistics for 802.1Q VLAN 1:
   441 packets, 85825 bytes input
   1028 packets, 69082 bytes output
Total statistics for 802.1Q VLAN 101:
   5173 packets, 510384 bytes input
   3042 packets, 369567 bytes output
Total statistics for 802.1Q VLAN 201:
   1012 packets, 119254 bytes input
   1018 packets, 120393 bytes output
Total statistics for 802.1Q VLAN 301:
   3163 packets, 265272 bytes input
   1011 packets, 120750 bytes output
Total statistics for 802.1Q VLAN 401:
   1012 packets, 119254 bytes input
   1010 packets, 119108 bytes output
```

Monitoring and Maintaining VLAN Subinterfaces

Use the following task to determine whether a VLAN is a native VLAN.

SUMMARY STEPS

1. **enable**
2. **show vlans**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show vlans** **Example:** `Router# show vlans`	Displays VLAN subinterfaces.

Monitoring and Maintaining VLAN Subinterfaces Example

The following is sample output from the **show vlans**command indicating a native VLAN and a bridged group:

```
Router# show vlans
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interface:   FastEthernet1/0/2
 This is configured as native Vlan for the following interface(s) :
FastEthernet1/0/2
   Protocols Configured:   Address: Received:        Transmitted:
Virtual LAN ID:  100 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interface:   FastEthernet1/0/2.1
   Protocols Configured:   Address: Received:        Transmitted:
        Bridging        Bridge Group 1 0                   0
```

The following is sample output from the **show vlans**command that shows the traffic count on Fast Ethernet subinterfaces:

```
Router# show vlans
Virtual LAN ID:  2 (IEEE 802.1Q Encapsulation)
    vLAN Trunk Interface:   FastEthernet5/0.1

    Protocols Configured:   Address:              Received:        Transmitted:
           IP              172.16.0.3                    16               92129

Virtual LAN ID:  3 (IEEE 802.1Q Encapsulation)

    vLAN Trunk Interface:   Ethernet6/0/1.1

    Protocols Configured:   Address:              Received:        Transmitted:
           IP              172.20.0.3                  1558                1521

Virtual LAN ID:  4 (Inter Switch Link Encapsulation)

    vLAN Trunk Interface:   FastEthernet5/0.2

    Protocols Configured:   Address:              Received:        Transmitted:
           IP              172.30.0.3                     0                   7
```

Configuration Examples for Configuring Routing Between VLANs

Single Range Configuration Example

The following example configures the Fast Ethernet subinterfaces within the range 5/1.1 and 5/1.4 and applies the following VLAN IDs to those subinterfaces:

Fast Ethernet5/1.1 = VLAN ID 301 (vlan-id)

Fast Ethernet5/1.2 = VLAN ID 302 (vlan-id = 301 + 2 - 1 = 302)

Fast Ethernet5/1.3 = VLAN ID 303 (vlan-id = 301 + 3 - 1 = 303)

Fast Ethernet5/1.4 = VLAN ID 304 (vlan-id = 301 + 4 - 1 = 304)

```
Router(config)# interface range fastethernet5/1.1 - fastethernet5/1.4

Router(config-if)# encapsulation dot1Q 301
Router(config-if)# no shutdown

Router(config-if)#
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.1, changed state to up
```

```
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.2, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.3, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.4, changed state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.1, changed
 state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.2, changed
 state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.3, changed
 state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.4, changed
 state to up
```

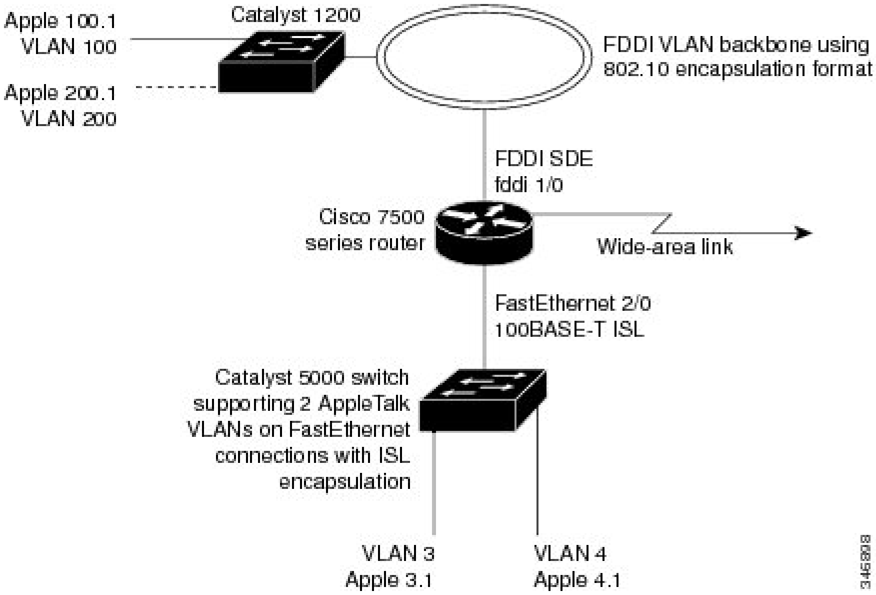
ISL Encapsulation Configuration Examples

This section provides the following configuration examples for each of the protocols described in this module:

AppleTalk Routing over ISL Configuration Example

The configuration example illustrated in the figure below shows AppleTalk being routed between different ISL and IEEE 802.10 VLAN encapsulating subinterfaces.

Figure 10: Routing AppleTalk over VLAN Encapsulations



As shown in the figure above, AppleTalk traffic is routed to and from switched VLAN domains 3, 4, 100, and 200 to any other AppleTalk routing interface. This example shows a sample configuration file for the Cisco 7500 series router with the commands entered to configure the network shown in the figure above.

Cisco 7500 Router Configuration

```
!
appletalk routing
interface Fddi 1/0.100
 encapsulation sde 100
 appletalk cable-range 100-100 100.2
 appletalk zone 100
```

```
!
interface Fddi 1/0.200
 encapsulation sde 200
 appletalk cable-range 200-200 200.2
 appletalk zone 200
!
interface FastEthernet 2/0.3
 encapsulation isl 3
 appletalk cable-range 3-3 3.2
 appletalk zone 3
!
interface FastEthernet 2/0.4
 encapsulation isl 4
 appletalk cable-range 4-4 4.2
 appletalk zone 4
!
```

Banyan VINES Routing over ISL Configuration Example

To configure routing of the Banyan VINES protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows Banyan VINES configured to be routed over an ISL trunk:

```
vines routing
interface fastethernet 0.1
 encapsulation isl 100
 vines metric 2
```

DECnet Routing over ISL Configuration Example

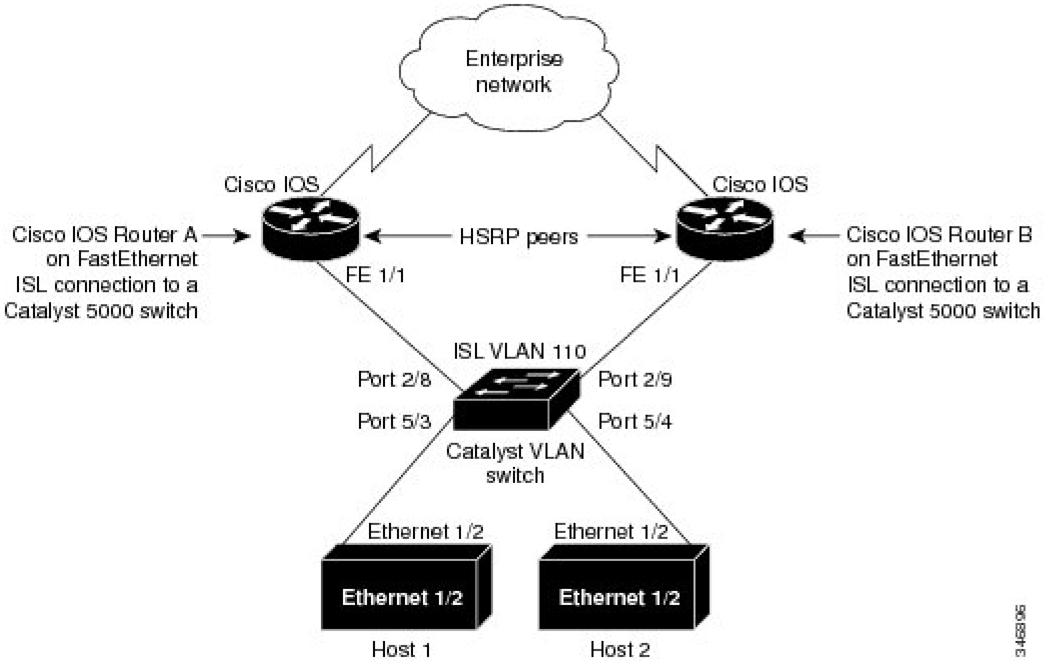
To configure routing the DECnet protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows DECnet configured to be routed over an ISL trunk:

```
decnet routing 2.1
interface fastethernet 1/0.1
 encapsulation isl 200
 decnet cost 4
```

HSRP over ISL Configuration Example

The configuration example shown in the figure below shows HSRP being used on two VLAN routers sending traffic to and from ISL VLANs through a Catalyst 5000 switch. Each router forwards its own traffic and acts as a standby for the other.

Figure 11: Hot Standby Router Protocol Sample Configuration



The topology shown in the figure above shows a Catalyst VLAN switch supporting Fast Ethernet connections to two routers running HSRP. Both routers are configured to route HSRP over ISLs.

The standby conditions are determined by the standby commands used in the configuration. Traffic from Host 1 is forwarded through Router A. Because the priority for the group is higher, Router A is the active router for Host 1. Because the priority for the group serviced by Host 2 is higher in Router B, traffic from Host 2 is forwarded through Router B, making Router B its active router.

In the configuration shown in the figure above, if the active router becomes unavailable, the standby router assumes active status for the additional traffic and automatically routes the traffic normally handled by the router that has become unavailable.

Host 1 Configuration

```
interface Ethernet 1/2
 ip address 10.1.1.25 255.255.255.0
 ip route 0.0.0.0 0.0.0.0 10.1.1.101
```

Host 2 Configuration

```
interface Ethernet 1/2
 ip address 10.1.1.27 255.255.255.0
 ip route 0.0.0.0 0.0.0.0 10.1.1.102
!
```

Router A Configuration

```
interface FastEthernet 1/1.110
 encapsulation isl 110
 ip address 10.1.1.2 255.255.255.0
 standby 1 ip 10.1.1.101
 standby 1 preempt
 standby 1 priority 105
 standby 2 ip 10.1.1.102
 standby 2 preempt
!
end
!
```

Router B Configuration

```
interface FastEthernet 1/1.110
 encapsulation isl 110
 ip address 10.1.1.3 255.255.255.0
 standby 1 ip 10.1.1.101
 standby 1 preempt
 standby 2 ip 10.1.1.102
 standby 2 preempt
 standby 2 priority 105
router igrp 1
!
network 10.1.0.0
network 10.2.0.0
!
```

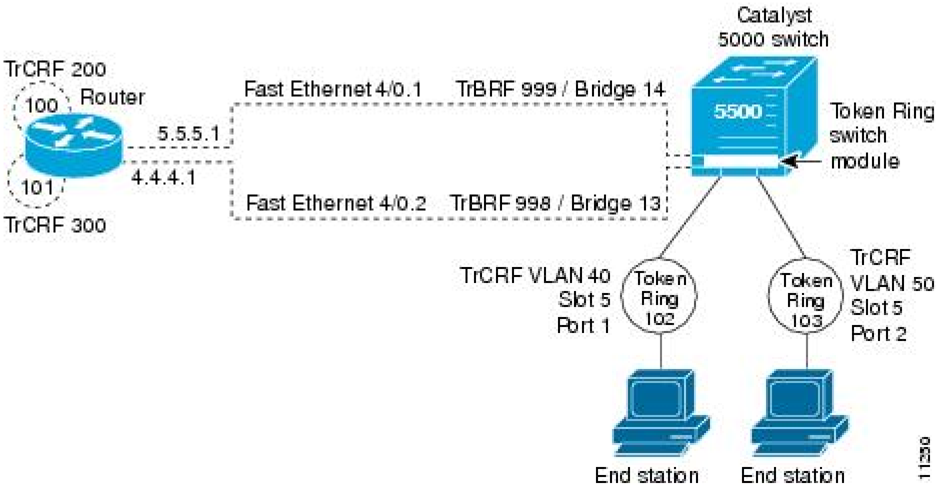
VLAN Switch Configuration

```
set vlan 110 5/4
set vlan 110 5/3
set trunk 2/8 110
set trunk 2/9 110
```

IP Routing with RIF Between TrBRF VLANs Example

The figure below shows IP routing with RIF between two TrBRF VLANs.

Figure 12: IP Routing with RIF Between TrBRF VLANs



The following is the configuration for the router:

```
interface FastEthernet4/0.1
 ip address 10.5.5.1 255.255.255.0
 encapsulation tr-isl trbrf-vlan 999 bridge-num 14
 multiring trcrf-vlan 200 ring 100
 multiring all
!
 interface FastEthernet4/0.2
 ip address 10.4.4.1 255.255.255.0
 encapsulation tr-isl trbrf-vlan 998 bridge-num 13
 multiring trcrf-vlan 300 ring 101
 multiring all
```

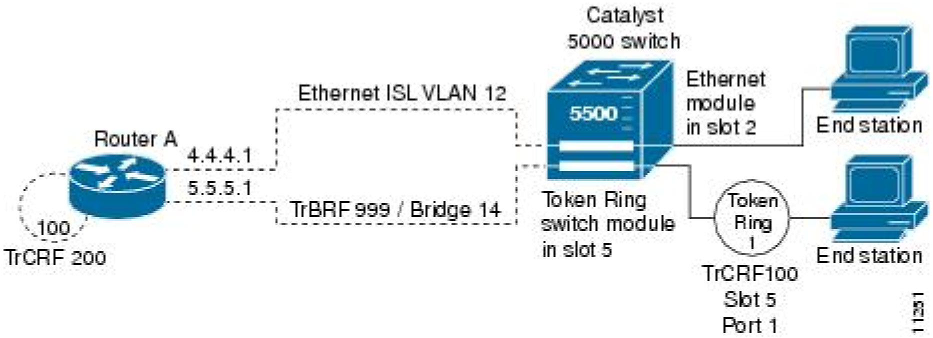
The following is the configuration for the Catalyst 5000 switch with the Token Ring switch module in slot 5. In this configuration, the Token Ring port 102 is assigned with TrCRF VLAN 40 and the Token Ring port 103 is assigned with TrCRF VLAN 50:

```
#vtp
set vtp domain trisl
set vtp mode server
set vtp v2 enable
#drip
set set tokenring reduction enable
set tokenring distrib-crf disable
#vlans
set vlan 999 name trbrf type trbrf bridge 0xe stp ieee
set vlan 200 name trcrf200 type trcrf parent 999 ring 0x64 mode srb
set vlan 40 name trcrf40 type trcrf parent 999 ring 0x66 mode srb
set vlan 998 name trbrf type trbrf bridge 0xd stp ieee
set vlan 300 name trcrf300 type trcrf parent 998 ring 0x65 mode srb
set vlan 50 name trcrf50 type trcrf parent 998 ring 0x67 mode srb
#add token port to trcrf 40
set vlan 40   5/1
#add token port to trcrf 50
set vlan 50   5/2
set trunk 1/2 on
```

IP Routing Between a TRISL VLAN and an Ethernet ISL VLAN Example

The figure below shows IP routing between a TRISL VLAN and an Ethernet ISL VLAN.

Figure 13: IP Routing Between a TRISL VLAN and an Ethernet ISL VLAN



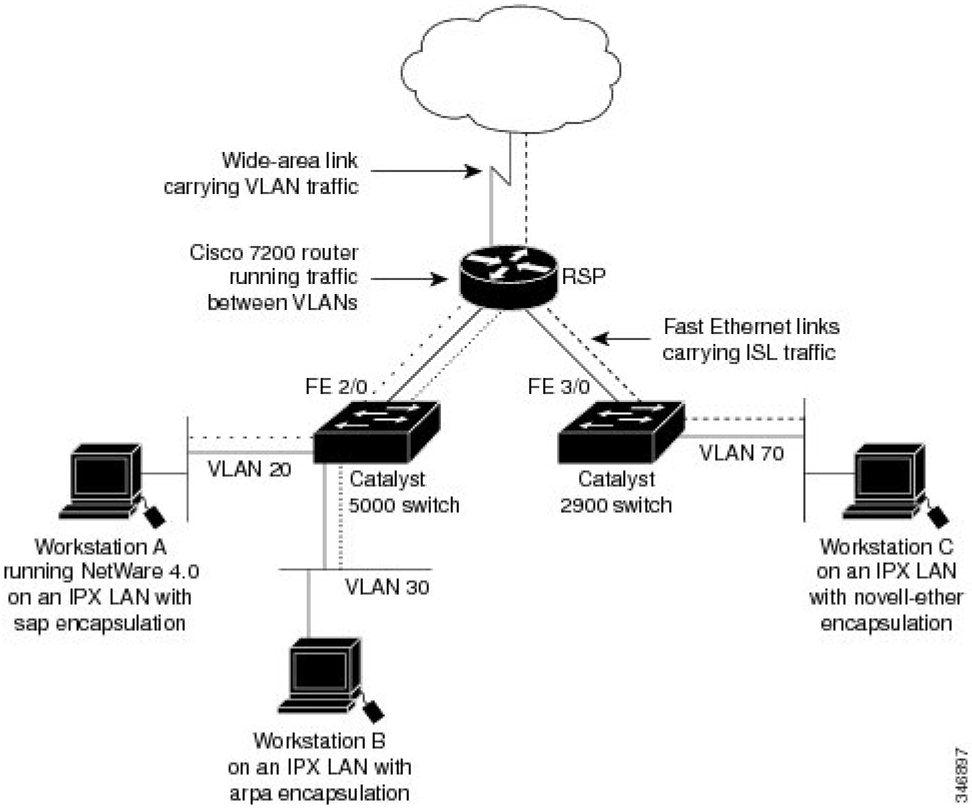
The following is the configuration for the router:

```
interface FastEthernet4/0.1
 ip address 10.5.5.1 255.255.255.0
 encapsulation tr-isl trbrf-vlan 999 bridge-num 14
 multiring trcrf-vlan 20 ring 100
 multiring all
!
interface FastEthernet4/0.2
 ip address 10.4.4.1 255.255.255.0
 encapsulation isl 12
```

IPX Routing over ISL Configuration Example

The figure below shows IPX interior encapsulations configured over ISL encapsulation in VLAN configurations. Note that three different IPX encapsulation formats are used. VLAN 20 uses SAP encapsulation, VLAN 30 uses ARPA, and VLAN 70 uses novell-ether encapsulation. Prior to the introduction of this feature, only the default encapsulation format, "novell-ether," was available for routing IPX over ISL links in VLANs.

Figure 14: Configurable IPX Encapsulations Routed over ISL in VLAN Configurations


Wide-area link carrying VLAN traffic
Cisco 7200 router running traffic between VLANs
RSP
Fast Ethernet links carrying ISL traffic
FE 2/0
FE 3/0
VLAN 20
Catalyst 5000 switch
Catalyst 2900 switch
VLAN 70
Workstation A running NetWare 4.0 on an IPX LAN with sap encapsulation
VLAN 30
Workstation C on an IPX LAN with novell-ether encapsulation
Workstation B on an IPX LAN with arpa encapsulation
346897

VLAN 20 Configuration

```
ipx routing
interface FastEthernet 2/0
 no shutdown
interface FastEthernet 2/0.20
 encapsulation isl 20
 ipx network 20 encapsulation sap
```

VLAN 30 Configuration

```
ipx routing
interface FastEthernet 2/0
 no shutdown
interface FastEthernet 2/0.30
 encapsulation isl 30
 ipx network 30 encapsulation arpa
```

VLAN 70 Configuration

```
ipx routing
interface FastEthernet 3/0
 no shutdown
```

```
interface Fast3/0.70
 encapsulation isl 70
 ipx network 70 encapsulation novell-ether
```

IPX Routing on FDDI Interfaces with SDE Example

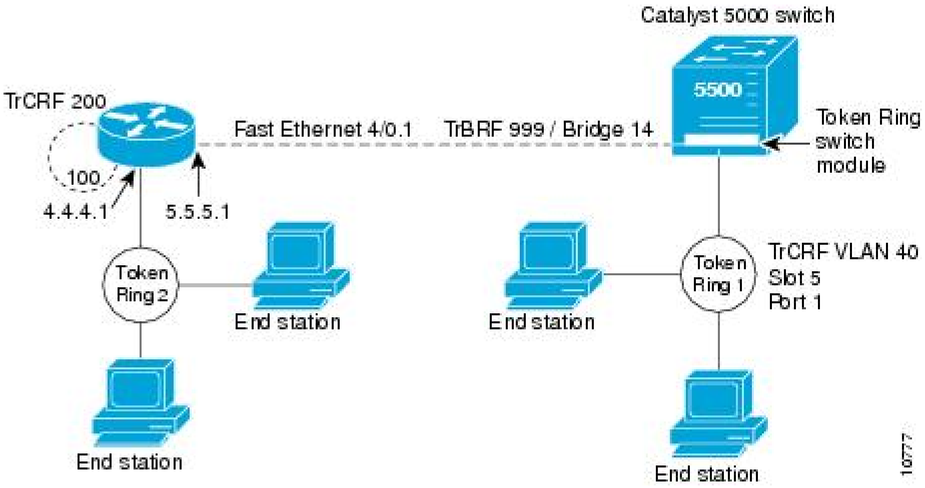
The following example enables IPX routing on FDDI interfaces 0.2 and 0.3 with SDE. On FDDI interface 0.2, the encapsulation type is SNAP. On FDDI interface 0.3, the encapsulation type is Novell's FDDI_RAW.

```
ipx routing
interface fddi 0.2 enc sde 2
 ipx network f02 encapsulation snap
interface fddi 0.3 enc sde 3
 ipx network f03 encapsulation novell-fddi
```

Routing with RIF Between a TRISL VLAN and a Token Ring Interface Example

The figure below shows routing with RIF between a TRISL VLAN and a Token Ring interface.

Figure 15: Routing with RIF Between a TRISL VLAN and a Token Ring Interface



The following is the configuration for the router:

```
source-bridge ring-group 100
!
interface TokenRing 3/1
 ip address 10.4.4.1 255.255.255.0
!
interface FastEthernet4/0.1
 ip address 10.5.5.1 255.255.255.0
 encapsulation tr-isl trbrf 999 bridge-num 14
 multiring trcrf-vlan 200 ring-group 100
 multiring all
```

The following is the configuration for the Catalyst 5000 switch with the Token Ring switch module in slot 5. In this configuration, the Token Ring port 1 is assigned to the TrCRF VLAN 40:

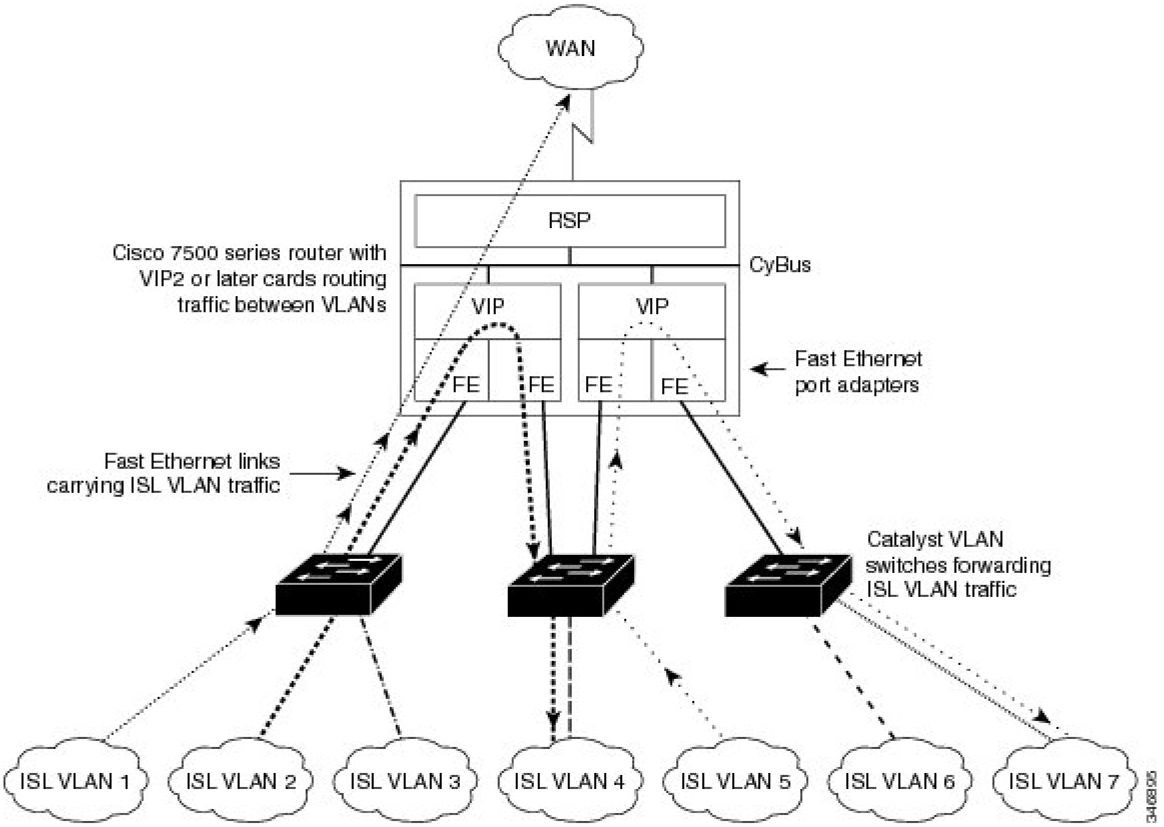
```
#vtp
set vtp domain trisl
set vtp mode server
set vtp v2 enable
```

```
#drip
set set tokenring reduction enable
set tokenring distrib-crf disable
#vlans
set vlan 999 name trbrf type trbrf bridge 0xe stp ieee
set vlan 200 name trcrf200 type trcrf parent 999 ring 0x64 mode srt
set vlan 40 name trcrf40 type trcrf parent 999 ring 0x1 mode srt
#add token port to trcrf 40
set vlan 40   5/1
set trunk 1/2 on
```

VIP Distributed Switching over ISL Configuration Example

The figure below shows a topology in which Catalyst VLAN switches are connected to routers forwarding traffic from a number of ISL VLANs. With the VIP distributed ISL capability in the Cisco 7500 series router, each VIP card can route ISL-encapsulated VLAN IP traffic. The inter-VLAN routing capacity is increased linearly by the packet-forwarding capability of each VIP card.

Figure 16: VIP Distributed ISL VLAN Traffic


WAN
RSP
Cisco 7500 series router with VIP2 or later cards routing traffic between VLANs
CyBus
VIP
VIP
FE
FE
FE
FE
Fast Ethernet port adapters
Fast Ethernet links carrying ISL VLAN traffic
Catalyst VLAN switches forwarding ISL VLAN traffic
ISL VLAN 1
ISL VLAN 2
ISL VLAN 3
ISL VLAN 4
ISL VLAN 5
ISL VLAN 6
ISL VLAN 7
346895

In the figure above, the VIP cards forward the traffic between ISL VLANs or any other routing interface. Traffic from any VLAN can be routed to any of the other VLANs, regardless of which VIP card receives the traffic.

These commands show the configuration for each of the VLANs shown in the figure above:

```
interface FastEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
 ip route-cache distributed
 full-duplex
interface FastEthernet1/0/0.1
 ip address 10.1.1.1 255.255.255.0
 encapsulation isl 1
interface FastEthernet1/0/0.2
 ip address 10.1.2.1 255.255.255.0
 encapsulation isl 2
interface FastEthernet1/0/0.3
 ip address 10.1.3.1 255.255.255.0
 encapsulation isl 3
interface FastEthernet1/1/0
 ip route-cache distributed
 full-duplex
interface FastEthernet1/1/0.1
 ip address 172.16.1.1 255.255.255.0
 encapsulation isl 4
interface Fast Ethernet 2/0/0
 ip address 10.1.1.1 255.255.255.0
 ip route-cache distributed
 full-duplex
interface FastEthernet2/0/0.5
 ip address 10.2.1.1 255.255.255.0
 encapsulation isl 5
interface FastEthernet2/1/0
 ip address 10.3.1.1 255.255.255.0
 ip route-cache distributed
 full-duplex
interface FastEthernet2/1/0.6
 ip address 10.4.6.1 255.255.255.0
 encapsulation isl 6
interface FastEthernet2/1/0.7
 ip address 10.4.7.1 255.255.255.0
 encapsulation isl 7
```

XNS Routing over ISL Configuration Example

To configure routing of the XNS protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows XNS configured to be routed over an ISL trunk:

```
xns routing 0123.4567.adcb
interface fastethernet 1/0.1
 encapsulation isl 100
 xns network 20
```

CLNS Routing over ISL Configuration Example

To configure routing of the CLNS protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows CLNS configured to be routed over an ISL trunk:

```
clns routing
interface fastethernet 1/0.1
 encapsulation isl 100
 clns enable
```

IS-IS Routing over ISL Configuration Example

To configure IS-IS routing over ISL trunks, you need to define ISL as the encapsulation type. This example shows IS-IS configured over an ISL trunk:

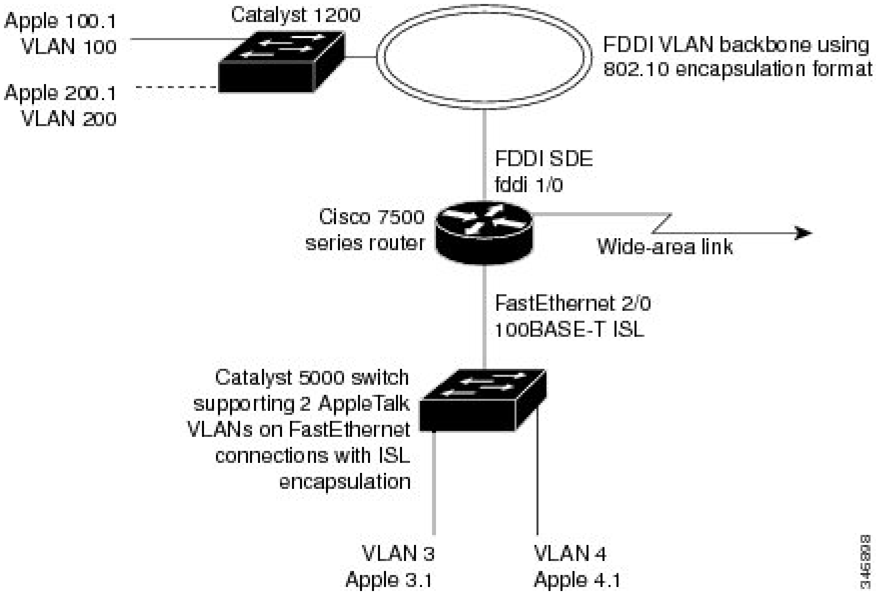
```
isis routing test-proc2
net 49.0001.0002.aaaa.aaaa.aaaa.00
```

```
interface fastethernet 2.0
 encapsulation isl 101
 clns router is-is test-proc2
```

Routing IEEE 802.10 Configuration Example

The figure below shows AppleTalk being routed between different ISL and IEEE 802.10 VLAN encapsulating subinterfaces.

Figure 17: Routing AppleTalk over VLAN encapsulations



As shown in the figure above, AppleTalk traffic is routed to and from switched VLAN domains 3, 4, 100, and 200 to any other AppleTalk routing interface. This example shows a sample configuration file for the Cisco 7500 series router with the commands entered to configure the network shown in the figure above.

Cisco 7500 Router Configuration

```
!
interface Fddi 1/0.100
 encapsulation sde 100
 appletalk cable-range 100-100 100.2
 appletalk zone 100
!
interface Fddi 1/0.200
 encapsulation sde 200
 appletalk cable-range 200-200 200.2
 appletalk zone 200
!
interface FastEthernet 2/0.3
 encapsulation isl 3
 appletalk cable-range 3-3 3.2
 appletalk zone 3
!
interface FastEthernet 2/0.4
 encapsulation isl 4
 appletalk cable-range 4-4 4.2
 appletalk zone 4
!
```

IEEE 802.1Q Encapsulation Configuration Examples

Configuration examples for each protocols are provided in the following sections:

Configuring AppleTalk over IEEE 802.1Q Example

This configuration example shows AppleTalk being routed on VLAN 100:

```
!
appletalk routing
!
interface fastethernet 4/1.100
  encapsulation dot1q 100
  appletalk cable-range 100-100 100.1
  appletalk zone eng
!
```

Configuring IP Routing over IEEE 802.1Q Example

This configuration example shows IP being routed on VLAN 101:

```
!
ip routing
!
interface fastethernet 4/1.101
  encapsulation dot1q 101
  ip addr 10.0.0.11 255.0.0.0
!
```

Configuring IPX Routing over IEEE 802.1Q Example

This configuration example shows IPX being routed on VLAN 102:

```
!
ipx routing
!
interface fastethernet 4/1.102
  encapsulation dot1q 102
  ipx network 100
!
```

VLAN 100 for Bridge Group 1 with Default VLAN1 Example

The following example configures VLAN 100 for bridge group 1 with a default VLAN1:

```
interface FastEthernet 4/1.100
encapsulation dot1q 1
bridge-group 1
```

VLAN 20 for Bridge Group 1 with Native VLAN Example

The following example configures VLAN 20 for bridge group 1 as a native VLAN:

```
interface FastEthernet 4/1.100
encapsulation dot1q 20 native
bridge-group 1
```

VLAN ISL or IEEE 802.1Q Routing Example

The following example configures VLAN ISL or IEEE 802.10 routing:

```
ipx routing
appletalk routing
!
interface Ethernet 1
ip address 10.1.1.1 255.255.255.0
appletalk cable-range 1-1 1.1
appletalk zone 1
ipx network 10 encapsulation snap
!
router igrp 1
network 10.1.0.0
!
end
!
#Catalyst5000
!
set VLAN 110 2/1
set VLAN 120 2/2
!
set trunk 1/1 110,120
# if 802.1Q, set trunk 1/1 nonegotiate 110, 120
!
end
!
ipx routing
appletalk routing
!
interface FastEthernet 1/1.110
encapsulation isl 110
!if 802.1Q, encapsulation dot1Q 110
ip address 10.1.1.2 255.255.255.0
appletalk cable-range 1.1 1.2
appletalk zone 1
ipx network 110 encapsulation snap
!
interface FastEthernet 1/1.120
encapsulation isl 120
!if 802.1Q, encapsulation dot1Q 120
ip address 10.2.1.2 255.255.255.0
appletalk cable-range 2-2 2.2
appletalk zone 2
ipx network 120 encapsulation snap
!
router igrp 1
network 10.1.0.0
network 10.2.1.0.0
!
end
!
ipx routing
appletalk routing
!
interface Ethernet 1
ip address 10.2.1.3 255.255.255.0
appletalk cable-range 2-2 2.3
appletalk zone 2
ipx network 120 encapsulation snap
!
router igrp 1
network 10.2.0.0
!
end
```

VLAN IEEE 802.1Q Bridging Example

The following examples configures IEEE 802.1Q bridging:

```
interface FastEthernet4/0
 no ip address
 no ip route-cache
 half-duplex
!
interface FastEthernet4/0.100
 encapsulation dot1Q 100
 no ip route-cache
 bridge-group 1
!
interface FastEthernet4/0.200
 encapsulation dot1Q 200 native
 no ip route-cache
 bridge-group 2
!
interface FastEthernet4/0.300
 encapsulation dot1Q 1
 no ip route-cache
 bridge-group 3
!
interface FastEthernet10/0
 no ip address
 no ip route-cache
 half-duplex
!
interface FastEthernet10/0.100
 encapsulation dot1Q 100
 no ip route-cache
 bridge-group 1
!
interface Ethernet11/3
 no ip address
 no ip route-cache
 bridge-group 2
!
interface Ethernet11/4
 no ip address
 no ip route-cache
 bridge-group 3
!
bridge 1 protocol ieee
bridge 2 protocol ieee
bridge 3 protocol ieee
```

VLAN IEEE 802.1Q IRB Example

The following examples configures IEEE 802.1Q integrated routing and bridging:

```
ip cef
appletalk routing
ipx routing 0060.2f27.5980
!
bridge irb
!
interface TokenRing3/1
```

```
 no ip address
 ring-speed 16
 bridge-group 2
!
interface FastEthernet4/0
 no ip address
 half-duplex
!
interface FastEthernet4/0.100
 encapsulation dot1Q 100
 bridge-group 1
!
interface FastEthernet4/0.200
 encapsulation dot1Q 200
 bridge-group 2
!
interface FastEthernet10/0
ip address 10.3.1.10 255.255.255.0
 half-duplex
 appletalk cable-range 200-200 200.10
 appletalk zone irb
 ipx network 200
!
interface Ethernet11/3
 no ip address
 bridge-group 1
!
interface BVI 1
 ip address 10.1.1.11 255.255.255.0
 appletalk cable-range 100-100 100.11
 appletalk zone bridging
 ipx network 100
!
router rip
 network 10.0.0.0
 network 10.3.0.0
!
bridge 1 protocol ieee
 bridge 1 route appletalk
 bridge 1 route ip
 bridge 1 route ipx
bridge 2 protocol ieee
!
```

Configuring IEEE 802.1Q-in-Q VLAN Tag Termination Example

Some ambiguous subinterfaces can use the **any** keyword for the inner VLAN ID specification. The **any** keyword represents any inner VLAN ID that is not explicitly configured on any other interface. In the following example, seven subinterfaces are configured with various outer and inner VLAN IDs.

Note The **any** keyword can be configured on only one subinterface of a specified physical interface and outer VLAN ID.

```
interface GigabitEthernet1/0/0.1
 encapsulation dot1q 100 second-dot1q 100
interface GigabitEthernet1/0/0.2
 encapsulation dot1q 100 second-dot1q 200
interface GigabitEthernet1/0/0.3
 encapsulation dot1q 100 second-dot1q 300-400,500-600
interface GigabitEthernet1/0/0.4
 encapsulation dot1q 100 second-dot1q any
interface GigabitEthernet1/0/0.5
 encapsulation dot1q 200 second-dot1q 50
interface GigabitEthernet1/0/0.6
```

```
 encapsulation dot1q 200 second-dot1q 1000-2000,3000-4000
interface GigabitEthernet1/0/0.7
 encapsulation dot1q 200 second-dot1q any
```

The table below shows which subinterfaces are mapped to different values of the outer and inner VLAN ID on Q-in-Q frames that come in on Gigabit Ethernet interface 1/0/0.

Table 6: Subinterfaces Mapped to Outer and Inner VLAN IDs for GE Interface 1/0/0

Outer VLAN ID	Inner VLAN ID	Subinterface mapped to
100	1 through 99	GigabitEthernet1/0/0.4
100	100	GigabitEthernet1/0/0.1
100	101 through 199	GigabitEthernet1/0/0.4
100	200	GigabitEthernet1/0/0.2
100	201 through 299	GigabitEthernet1/0/0.4
100	300 through 400	GigabitEthernet1/0/0.3
100	401 through 499	GigabitEthernet1/0/0.4
100	500 through 600	GigabitEthernet1/0/0.3
100	601 through 4095	GigabitEthernet1/0/0.4
200	1 through 49	GigabitEthernet1/0/0.7
200	50	GigabitEthernet1/0/0.5
200	51 through 999	GigabitEthernet1/0/0.7
200	1000 through 2000	GigabitEthernet1/0/0.6
200	2001 through 2999	GigabitEthernet1/0/0.7
200	3000 through 4000	GigabitEthernet1/0/0.6
200	4001 through 4095	GigabitEthernet1/0/0.7

A new subinterface is now configured:

```
interface GigabitEthernet1/0/0.8
 encapsulation dot1q 200 second-dot1q 200-600,900-999
```

The table below shows the changes made to the table for the outer VLAN ID of 200. Notice that subinterface 1/0/0.7 configured with the **any** keyword now has new inner VLAN ID mappings.

Table 7: Subinterfaces Mapped to Outer and Inner VLAN IDs for GE Interface 1/0/0--Changes Resulting from Configuring GE Subinterface 1/0/0.8

Outer VLAN ID	Inner VLAN ID	Subinterface mapped to
200	1 through 49	GigabitEthernet1/0/0.7
200	50	GigabitEthernet1/0/0.5
200	51 through 199	GigabitEthernet1/0/0.7
200	200 through 600	GigabitEthernet1/0/0.8
200	601 through 899	GigabitEthernet1/0/0.7
200	900 through 999	GigabitEthernet1/0/0.8
200	1000 through 2000	GigabitEthernet1/0/0.6
200	2001 through 2999	GigabitEthernet1/0/0.7
200	3000 through 4000	GigabitEthernet1/0/0.6
200	4001 through 4095	GigabitEthernet1/0/0.7

Additional References

The following sections provide references related to configuring a VLAN range.

Related Documents

Related Topic	Document Title
IP LAN switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS LAN Switching Command Reference
SNMP	Configuring SNMP Support module in the *Cisco IOS Network Management Configuration Guide*
HSRP	Configuring HSRP" module in the *Cisco IOS IP Application Services Configuration Guide*
Encapsulation types and corresponding framing types	Configuring Novell IPX module in the *Cisco IOS Novell IPX Configuration Guide*
AppleTalk	Configuring AppleTalk module in the *Cisco IOS AppleTalk Configuration Guide*

Standards

Standard	Title
IEEE 802.10 standard	802.10 Virtual LANs

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Routing Between VLANs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 8: Feature Information for Routing Between VLANs

Feature Name	Releases	Feature Information
IEEE 802.1Q-in-Q VLAN Tag Termination	12.0(28)S, 12.3(7)(X17) 12.0(32)S1, 12.2(31)SB 12.3(7)T 12.3((7)XI1	Encapsulating IEEE 802.1Q VLAN tags within 802.1Q enables service providers to use a single VLAN to support customers who have multiple VLANs. The IEEE 802.1Q-in-Q VLAN Tag Termination feature on the subinterface level preserves VLAN IDs and keeps traffic in different customer VLANs segregated.
Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T Cisco IOS XE 3.8(S) Cisco IOS XE 3.9(S)	The IEEE 802.1Q protocol is used to interconnect multiple switches and routers, and for defining VLAN topologies. The IEEE 802.1Q standard is extremely restrictive to untagged frames. The standard provides only a per-port VLANs solution for untagged frames. For example, assigning untagged frames to VLANs takes into consideration only the port from which they have been received. Each port has a parameter called a *permanent virtual identification* (Native VLAN) that specifies the VLAN assigned to receive untagged frames. In Cisco IOS XE Release 3.8(S), support was added for the Cisco ISR 4400 Series Routers. In Cisco IOS XE Release 3.9(S), support was added for the Cisco CSR 1000V Series Routers.

Feature Name	Releases	Feature Information
Configuring Routing Between VLANs with Inter-Switch Link Encapsulation	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T	ISL is a Cisco protocol for interconnecting multiple switches and maintaining VLAN information as traffic goes between switches. ISL provides VLAN capabilities while maintaining full wire speed performance on Fast Ethernet links in full- or half-duplex mode. ISL operates in a point-to-point environment and will support up to 1000 VLANs. You can define virtually as many logical networks as are necessary for your environment.
Configuring Routing Between VLANs with IEEE 802.10 Encapsulation	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T	AppleTalk can be routed over VLAN subinterfaces using the ISL or IEEE 802.10 VLANs feature that provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

Feature Name	Releases	Feature Information
VLAN Range	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T	Using the VLAN Range feature, you can group VLAN subinterfaces together so that any command entered in a group applies to every subinterface within the group. This capability simplifies configurations and reduces command parsing. In Cisco IOS Release 12.0(7)XE, the **interface range** command was introduced. The **interface range** command was integrated into Cisco IOS Release 12.1(5)T. In Cisco IOS Release 12.2(2)DD, the **interface range** command was expanded to enable configuration of subinterfaces. The **interface range** command was integrated into Cisco IOS Release 12.2(4)B. The VLAN Range feature was integrated into Cisco IOS Release 12.2(8)T. This VLAN Range feature was integrated into Cisco IOS Release 12.2(13)T.
256+ VLANS	12.1(2)E, 12.2(8)T Cisco IOS XE 3.8(S) Cisco IOS XE 3.9(S)	The 256+ VLAN feature enables a device to route more than 256 VLAN interfaces. This feature requires the MSFC2. The routed VLAN interfaces can be chosen from any of the VLANs supported on the device. Catalyst switches can support up to 4096 VLANs. If MSFC is used, up to 256 VLANs can be routed, but this can be selected from any VLANs supported on the device. In Cisco IOS XE Release 3.8(S), support was added for the Cisco ISR 4400 Series Routers. In Cisco IOS XE Release 3.9(S), support was added for the Cisco CSR 1000V Series Routers.